Application of *SPH* numerical procedures to hydraulic problems

Enrico Vitanza

Department of Engineering

University of Palermo

A thesis submitted for the degree of

Doctor of Philosophy

10/03/2014

I would like to dedicate this thesis to all the people who supported me, during these past three years. A special thanks goes to Professors Enrico Napoli and Diego Molteni whose fundamental collaboration made this work possible.

Abstract

Smoothed Particle Hydrodynamics (SPH) is a widely used numerical procedure in CFD. Born in the seventies in order to simulate astrophysical phenomena of gas dynamics, it was stated later in other areas of science and engineering, from solid to fluid mechanics. Its Lagrangian nature and capability to simulate complex free surface flows, shocks and large fluids deformations, certify it as one of the most appreciated among the mesh-free methods. This work is focused in developing and testing the SPH algorithm in different flow simulations, carrying out results that are in good agreement with analytical ones and, when available, with experimental data. In the following, the exploit of a 2D/3D fluid numerical code implemented in its two variants, weakly compressible (WCSPH) and truly incompressible (ISPH) formulation, will be described in detail. Basically, dealing with fluids, the difference between the two versions lies on the pressure calculation: while WCSPH uses an equation of state (EOS) to obtain a pressure value in each point of computational domain, ISPH exploits a pressure Poisson equation using a projection-based method. To show the validity of the developed code, the results of various standard test cases, mostly free-surface flows, will be shown. More precisely, the 2D/3D dam break, landslide run-up, shear cavity and surface tension effects, along with issues related to the boundary conditions, will be taken in consideration. Here, the SPH meshless procedure is proposed as a viable alternative to the fixed-grid based approach, showing how the weakness of the latter, become strengths for the first. To give further confirmation of the above, will be shown an entirely new approach based on the SPH method, to the problem of non-reflecting boundary conditions on a finite domain. Specifically, it will be shown that the Lagrangian technique provides results as good as those by the Eulerian one, but with advantages of an easy implementation and the absence of problems related to the construction of a calculation grid.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational fluid-dynamics (CFD) has made great strides in recent years, thanks to growing computer capability and more and more sophisticated calculation methods. Though this is beyond the aim of this thesis, it's worth summarizing the main numerical techniques to solve the problems related to fluids. The most important classification of these procedures concerns the type of approach which flows of fluid are described with, that is, Eulerian or Lagrangian approach. From the Eulerian point of view, a flow quantity is studied, by varying the time, in a fixed point of the space, while from Lagrangian one, the same quantity is studied following the fluid during its motion, that is, the observer is in a reference system fixed with the flow. Historically in CFD, Eulerian (and Lagrangian) approaches are based on *grids* or *meshes*. Numerical Eulerian procedures such as the finite difference methods (FDM) (Morton and Mayers [2005]) and the finite volume methods (FVM) (Toro [1999]) are commonly known as *fixed* grid-based methods, while the Lagrangian finite element methods (FEM) (Baker [1985]) relies on grids attached on the modeled medium. They all have been applied over decades in almost every area of fluid dynamics to describe, study and solve the set of differential or partial differential equations related to physical phenomena. In the grid-based numerical models, a continuos domain is divided into smaller discrete sub-domains through a *meshing* process where the individual nodes are related each other by a topological map, that is, the mesh or grid. This kind of discretization results in elements in FEM, grids in FDM and cells in FVM. After these preliminary operations on the computational domain, is possible to convert the governing equations to a set of algebraic equations with nodal unknowns for the field variables. All of these grid-based numerical method have had a great success and application in the field of computational continuum mechanics due to their flexibility and accuracy, such that they are currently the dominant methods in numerical simulations. Despite their success and wide range of applicability, these methods suffer several drawbacks. The use of a cal-

culation grid is both an advantage and a disadvantage in dealing with problems with free surface, deformable boundaries, moving interface (for FDM and FVM) and extremely large deformation (for FEM): excessive domain deformation requires cpu-expensive remeshing algorithm and instabilities caused by a large grid distortion may arise. Problems with complicated geometry require a fine grid production that, in addition in being a not easy task and introducing solution inaccuracies, is a time consuming process even for the last generation machines. Moreover, for free surface problems, the Eulerian procedures need to be coupled with a mathematical treatment such as the formulation of an advection scheme to keep track and locate free surface areas. For an example of how this can be done, see the volume of fluid (VOF) method by Hirt and Nichols [1981]. However the discretization of the advection terms in VOF-based models, yields issues related to an unwanted numerical diffusion.
In recent decades, more attention has been focused on Lagrangian *meshfree* numerical methods, precisely because of their natural flexibility in managing problems where the Eulerian grid methods lacked. The winning idea in this type of procedure is to solve the integral or partial differential equations related to physical problem (along with certain boundary conditions), using a set of arbitrarily distributed nodes or particles. For a deeper knowledge on meshless methods see Belytschko et al. [1996] et al., Liu [2002], Liu and Gu [2005]. This thesis will focus on a development and application of a distinct meshfree particle numerical procedure, which is a very powerful and versatile method for solving CFD problems governed by the Navier-Stokes equations. The just mentioned procedure is called *Smoothed Particle Hydrodynamics* (SPH). This is the oldest meshfree particle numerical method based on Lagrangian dynamics, independently introduced by Gingold and Monaghan [1977] and Lucy [1977] for modeling complex and highly non-symmetrical fluid-dynamical phenomena in astrophysics. Its application was subsequently extended to problems of continuum solid and fluid mechanics. The basic idea is to replace the fluid with a set of N particles without a fixed connection, typically with constant masses, arranged in such a way to simulate a continuos medium, rather than individual points in space. Such a prior condition is due to the need to calculate the density of a fluid, which is a continuos variable, to compute hydrodynamical forces acting on a particle. In other words, in SPH formulation the equations governing a fluid are transformed in fluxes and forces among the particles. As pointed out, the meshless nature and the relative easy-implementation of this method have attracted great interest in the field of fluid numerical simulation, especially for free surface flows. An immediate benefit with respect to Eulerian grid-based methods it that SPH, using a Lagrangian approach, allows the advection terms to be calculated without numerical diffusion. Secondly, it does not suffer from the mesh distortion problems that limits the simulation of large mass deformation in a structured

mesh. Finally, because of its fully Lagrangian nature, keeps automatically track of the shape of the fluid free surface, even in presence of excessive fragmentation. This work, in particular, is focused on the standard basic SPH formulation applied to fluids and one of its modification, the *Incompressible Smoothed Particle Hydrodynamics* (ISPH) that allows, as the name suggests, to deal with fluids satisfying the incompressibility condition exactly. The most remarkable differences between the two mentioned methods, when treating a fluid like water, is that the former forces the incompressibility constraint by a suitable equation of state (Batchelor [1967], Monaghan [1994]) that relates pressure and density, while the latter solves a partial differential equation for pressure (*Poisson* equation). Unfortunately, each of these approaches has its own drawbacks: standard SPH with equation of state to solve the pressure does not fulfill exactly the incompressibility condition, making the fluid 'quasi-incompressible' (density fluctuations are in the order of $1 \sim 2$ %); for this reason it is called *Weakly Compressible Smoothed Particle Hydrodynamics* (WCSPH). The use of a stiff equation of state (Becker and Teschner [2007], Becker et al. [2009] et al.) generates acoustic waves traveling throughout the medium, thus perturbating the fluid dynamics. Furthermore, the choice of parameters adopted in the above mentioned equation imposes several time-step restrictions to satisfy the Courant-Friedrichs-Levy condition (CFL). On the other hand, the ISPH approach, while eliminating the issue of acoustic waves and the use of time steps prohibitively small, has a computational cost much higher than WCSPH. This fully Lagrangian scheme enforces incompressibility in a very like way to eulerian ones (Chorin [1968], Bell et al. [1989] et al., Enright et al. [2002] et al.) firstly integrating the intermediate velocity field without considering the pressure contribution, secondly projecting this velocity field onto a divergence-free space to fulfill incompressibility trough a pressure Poisson equation (Cummins and Rudman [1999]). Using this procedure, the solution of an elliptical problem that arises in the projection step, increases the cpu load very much. The present work has been divided in two main phases: the first one was devoted to the development *from scratch* of a SPH code including the two aforementioned variants (WCSPH, ISPH). The second one was dedicated to test the code to various hydraulic problem, such as the dam break, landslide rising, surface tension effects and the open boundaries related problem. In the latter case, a novel SPH application, in the Lagrangian framework, based on the Perfectly Matched Layer (PML) approach to treat non reflecting boundaries, will be shown. All the results obtained showed the same accuracy when compared to those produced by grid-based codes, as well the SPH ability to reproduce with good approximation analytical and, when available, experimental results.

1.1 Thesis arrangement

This thesis is subdivided as follows: in **Chapter 2** will be provided the theoretical background of the SPH methodology along with the implementation techniques for numerical simulations of weakly and full incompressible fluids. In particular, it will be highlighted the advantages and disadvantages in using either approach. **Chapter 3**, will focus on the results carried out in various test cases, as well as the comparison between the SPH and the Eulerian FVM, in order to validate the developed codes. In **Chapter 4** will be addressed the discussion regarding the implementation of absorbing layers used as boundary conditions (BC) in shallow water models with finite domain. After a brief review on the derivation of the governing equations, a brand new approach for the construction of these BC, based on the Lagrangian SPH technique, will be presented. **Chapter 5** will therefore be devoted to the presentation of the obtained results. Finally, **Chapter 6** will be reserved for conclusions.

Chapter 2

The SPH methodology

2.1 Foundations

In this section, the basic concepts of SPH numerical method will be introduced. Further details can be found in Monaghan [1992] and Benz [1989], in which the two authors use slightly different approaches to get substantially the same equations. Before considering the merits of the SPH methodology, will be carried out some general considerations on the main features of the numerical codes for solving systems of PDEs. The most important problem that these codes have to face is given by the domain discretization, that must be introduced to approximate, by algebraic calculations, the values of the derivatives appearing in the equations. The most common approach to this problem is that of finite difference numerical codes. To fix ideas, consider a two-dimensional case where a system of differential equations whose unknowns are functions of two spatial variable, x and y, and time t. In such a situation, for calculating the spatial derivative of the unknown functions, is necessary to create in the portion of the xy plane (that constitutes the domain of the unknowns) a *grid* aimed to define the discrete set of points on which will be performed the calculation of the difference quotients of the aforementioned functions. These quotients, when the size of the grid cells tend to zero (infinitely fine grid), become the derivatives of the unknown functions themselves. In the finite differences codes, during the system time evolution, the grid is held fixed. The critical parameter of this approximation method is the size of the grid cells. With a smaller size, one can gain accuracy, but the *cpu*-cost is expensive. On the other hand, increasing the size the opposite evidently occurs. It is therefore necessary to find a compromise between the need for accuracy and the speed of calculation, by defining a suitable size for the grid cells. So, the definition of a good value of the "cell size", is a quite common issue in developing numerical codes to solve system of PDEs.
The SPH algorithm is a numerical method based on a Lagrangian description of

the fluid motion. Recall that the term 'Lagrangian' means a description in which each portion of the fluid is followed along its motion. In other words, temporal evolution of the physical quantities of interest (such as the velocity or pressure field) are not provided in fixed points of the computational domain, but in the instantaneous position of the fluid portion considered. This means that the 'grid' required for the derivative calculation is not fixed, but distorted by the motion of the various portions in which the fluid is divided, during the temporal evolution.

2.1.1 SPH representation of a function: integral and *particle* approximation

In the SPH formalism, the portions which the fluid is divided in, are usually called "*particles*". The heart of SPH algorithm consists of an interpolating function, which is used to estimate the physical quantities and their derivatives in each spatial point of a disordered system of particles. Now, let's consider the identity:

$$A(\mathbf{r}) = \int_{\Omega} A(\mathbf{r}')\delta(\mathbf{r} - \mathbf{r}')d\mathbf{r}', \tag{2.1}$$

where A is a function of the three-dimensional position vector $\mathbf{r}$, and $\delta(\mathbf{r} - \mathbf{r}')$ is the Dirac delta function given by:

$$\delta(\mathbf{r} - \mathbf{r}') = \begin{cases} 1 & if \qquad \mathbf{r} = \mathbf{r}' \\ 0 & if \qquad \mathbf{r} \neq \mathbf{r}'. \end{cases} \tag{2.2}$$

Equation 2.1 means that a function $A(\mathbf{r})$, continuos in a domain of integration Ω, can be represented in an exact integral form. If the Dirac function is replaced with an interpolating kernel $W(\mathbf{r}, h)$, we obtain an interpolating integral of $A(\mathbf{r})$:

$$A_I(\mathbf{r}) \doteq \int_{\Omega} A(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}', \tag{2.3}$$

with the kernel having the following properties:

$$\int_{\Omega} W(r - r', h)dr' = 1, \tag{2.4}$$

$$\lim_{h \to 0} W(\mathbf{r} - \mathbf{r}', h) = \delta(\mathbf{r} - \mathbf{r}'), \tag{2.5}$$

$$W(\mathbf{r}, h) = W(-\mathbf{r}, h), \tag{2.6}$$

and

$$W(\mathbf{r} - \mathbf{r}', h) = 0 \qquad if \qquad |\mathbf{r} - \mathbf{r}'| > \kappa h. \tag{2.7}$$

Eq. 2.3 is called *kernel approximation*, eqs. 2.4, 2.5 and 2.7 represent respectively the *normalization*, *Delta function property* and *compact* condition. The first condition implies that integration of the function W, over the definition domain Ω, yields the unity. The second condition says that, when the parameter h approaches zero, W has the same properties as the Dirac function. Finally, the third condition defines, through the constant κ, the interval in which W has non-zero values. This interval represents the radius of the definition domain Ω which is called *support domain*. It can be demonstrated that if conditions 2.4 and 2.6 are fulfilled, then the kernel approximation of a function 2.3 has an accuracy of $O(h^2)$ (Hernquist and Katz [1989], Monaghan [1992], Fulk [1994]). The kernel function changes the characteristic of the particle: from a point mass to a spatially extended. More specifically, W is a "*bell shaped*", positive definite continuos function with continuos first derivative, used to spread the mass (and other properties) of a particle throughout the space and is normalized in a such way to preserve the physical quantities of the system. A simple mathematical application that meets these requirements is the Gaussian function:

$$W_{Gauss}(r,h) = \frac{1}{h^{\nu}\pi^{\frac{\nu}{2}}} exp(-\frac{r^2}{h^2}), \tag{2.8}$$

where ν is the number of spatial dimension, r is the distance from the particle and h is a variable of proportionality with the dimensions of a length. The variable h, already seen in the above equations, is what is called "*smoothing length*" since it controls the degree of particle 'diffusion' in space (Fig.2.1).

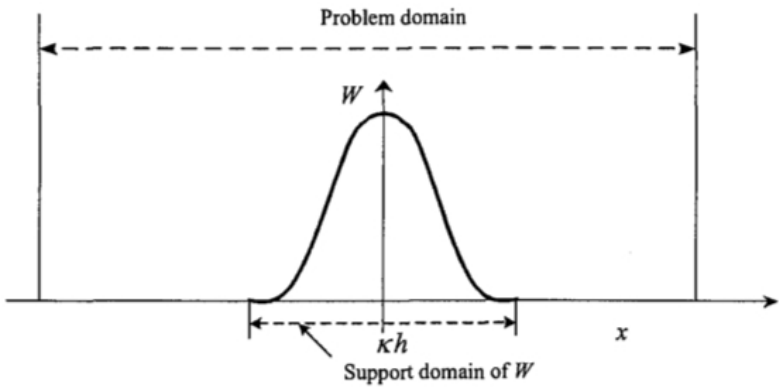


Figure 2.1: One dimensional kernel support domain embedded in the problem domain.

As stated before, the integral interpolating $A_i(\mathbf{r})$ can be thought of as a 'smoothed' version of the original function $A(\mathbf{r})$. The next step to final SPH

equations, called "*particle approximation*", is to approximate the continuos integral in 2.3 with an interpolating summation on a number j of points (particles) in the finite support domain of the kernel (Fig. 2.2). Replacing the infinitesimal volume element $d\mathbf{r}'$ at location of particle j with its finite volume $\triangle V_j$ and exploiting the relation:

$$m_j = \rho_j \triangle V_j, \tag{2.9}$$

where ρ_j is particle density, m_j its mass and $j = 1, 2, ..., N$ is the total number of particles in the support domain, we have:

$$A_S(\mathbf{r}_i) \doteq \sum_j^N \frac{m_j}{\rho_j} A(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h). \tag{2.10}$$

The above equation indicates that, given a domain containing a total number of particles N_{TOT}, the value of the quantity A, at position of each particle $i = 1, 2, ..., N_{TOT}$, is obtainable via an interpolating summation of the values of the same quantity over the $j = 1, 2, ..., N$ particles surrounding i. It's worth noting, that this further approximation introduces two key features:

- the sum is extended *only* to the particles that are in the support domain.
- the physical quantities, related to the particle, mass and density appear into the equations.

This makes it much easier to apply the SPH formalism to fluid dynamics problems, where the calculation of the aforementioned physical quantities is essential and the reduction of the number of "nodes" where to perform the calculation is crucial from the computational point of view.
Because A, appearing in eqs. 2.1, 2.3, 2.10 is a generic function, it can be replaced with the particle density ρ to give the *density summation approach* equation:

$$\rho(\mathbf{r}_i) = \sum_j^N m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{2.11}$$

2.1.2 SPH approximation of the derivative of a function: gradient and divergence operators

Applying the divergence operator to the approximated function $A_i(\mathbf{r})$ in eq. 2.3, we obtain:

$$\nabla \cdot A_I(\mathbf{r}) \doteq \int_\Omega \left[\nabla \cdot A(\mathbf{r}')\right] W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}', \tag{2.12}$$

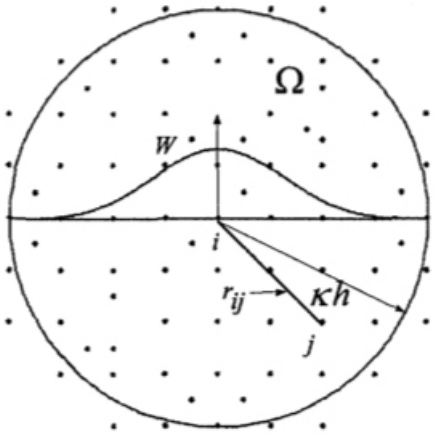


Figure 2.2: Sketch of the kernel support domain Ω for a particle i. Only particles j within a radius κh are considered for the interactions.

where the spatial derivative is taken respect to $\mathbf{r}'$. Since the following relation holds:

$$[\nabla \cdot A(\mathbf{r}')]\, W(\mathbf{r}-\mathbf{r}', h) = \nabla \cdot [A(\mathbf{r}')W(\mathbf{r}-\mathbf{r}', h)] - A(\mathbf{r}') \cdot \nabla W(\mathbf{r}-\mathbf{r}', h), \quad (2.13)$$

eq. 2.12 can be written as:

$$\nabla \cdot A_I(\mathbf{r}) \doteq \int_\Omega \nabla \cdot [A(\mathbf{r}')W(\mathbf{r}-\mathbf{r}', h)]\, d\mathbf{r}' - \int_\Omega A(\mathbf{r}') \cdot \nabla W(\mathbf{r}-\mathbf{r}', h) d\mathbf{r}'. \quad (2.14)$$

Now, using the divergence theorem in the first integral to the right hand side (RHS) of 2.14, we have:

$$\nabla \cdot A_I(\mathbf{r}) \doteq \int_\Sigma \nabla \cdot [A(\mathbf{r}')W(\mathbf{r}-\mathbf{r}', h)] \cdot \hat{\mathbf{n}} d\Sigma - \int_\Omega A(\mathbf{r}') \cdot \nabla W(\mathbf{r}-\mathbf{r}', h) d\mathbf{r}', \quad (2.15)$$

where Σ is the surface of the support domain Ω and $\hat{\mathbf{n}}$ is the unit vector normal to Σ (Fig. 2.3).

If the support domain is totally embedded into the problem domain (Figs. 2.1, 2.3), since the kernel W is zero on the boundaries (it is said having a compact

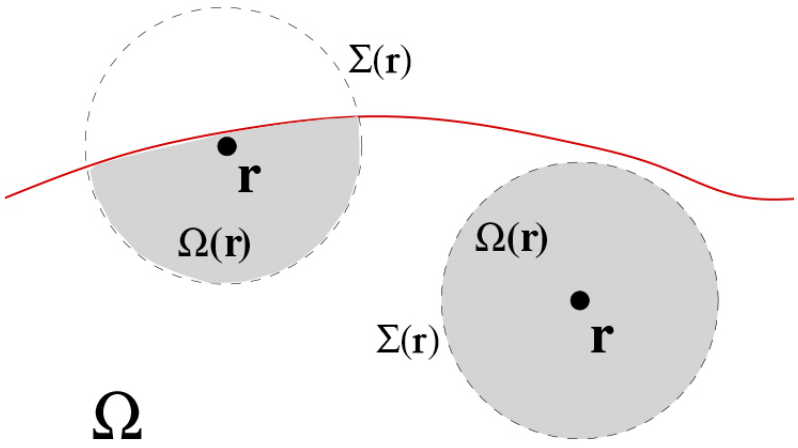


Figure 2.3: Volume (Ω) and surface (Σ) integrals over the support domains. Due to kernel truncation, the surface integral to the left is not identically zero.

support), the surface integral is also equal to zero. So, in this case, eq. 2.15 reduces to:

$$\nabla \cdot A_I(\mathbf{r}) \doteq -\int_\Omega A(\mathbf{r}') \cdot \nabla W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}'. \tag{2.16}$$

If, on the other hands, the support domain intercepts the edges of the problem domain (Figs. 2.4, 2.3), then the smoothing kernel is truncated and the surface integral is no longer zero. To deal with such situations, some strategies are used to create boundary conditions that can balance, fully or partly, the lack of accuracy. The implementation of the BC, will be discussed later, and recovered in Chapter 4.

As already seen in 2.10 and 2.11, switching from the continuos kernel approximation to the discrete SPH particle approximation is quite straightforward. For the *divergence*, we have:

$$\nabla \cdot A_S(\mathbf{r}_i) \doteq -\sum_j^N \frac{m_j}{\rho_j} A(\mathbf{r}_j) \cdot \nabla W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{2.17}$$

with the kernel gradient ∇W taken with respect to the particle j. Now, since the kernel W is an even function and its gradient is an odd one (Figs.2.5 (a),(b)), taking the derivative with respect to the particle i, the minus sign, outside the

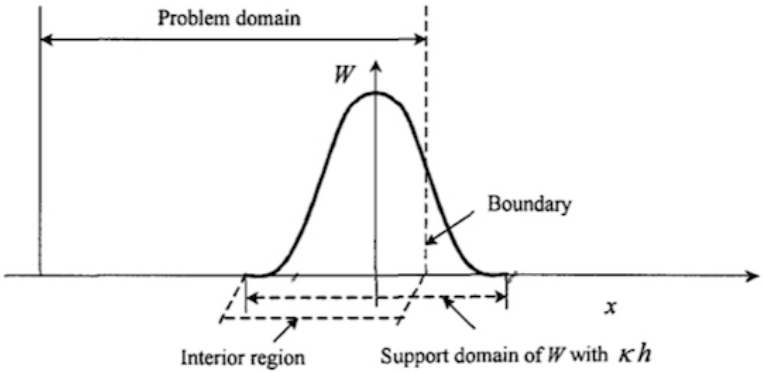


Figure 2.4: Overlapping support and problem domains. This occurrence produces a kernel truncation and makes the surface integral, in the RHS of 2.15, nonzero.

summation, can be eliminated:

$$\nabla \cdot A_S(\mathbf{r}_i) \doteq \sum_j^N \frac{m_j}{\rho_j} A(\mathbf{r}_j) \cdot \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{2.18}$$

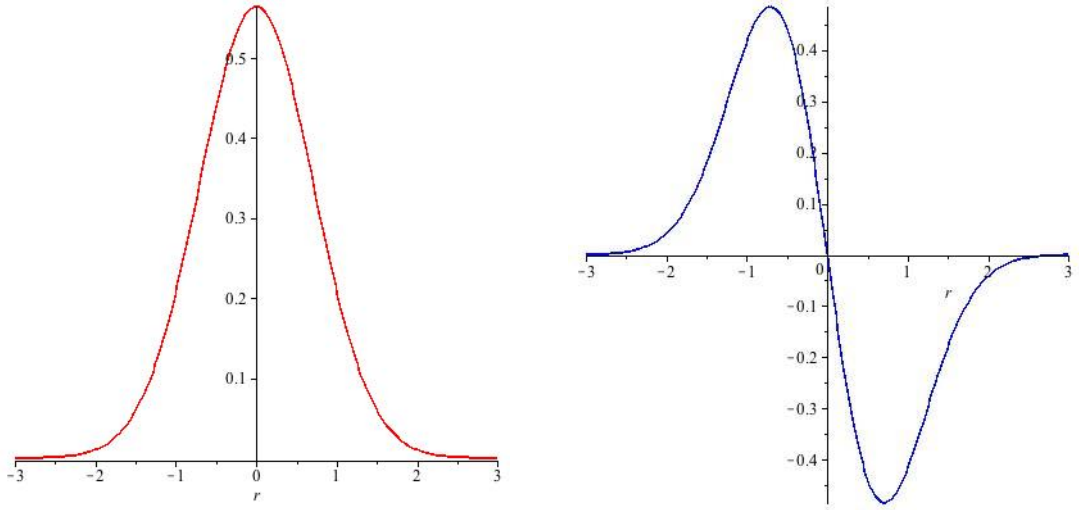
and, similarly, for the *gradient*:

$$\nabla A_S(\mathbf{r}_i) \doteq \sum_j^N \frac{m_j}{\rho_j} A(\mathbf{r}_j) \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{2.19}$$

where the notation ∇_i indicates the derivative with respect to the index i and the function A_S clearly represents, a vector in 2.18 and a scalar in 2.19.

Although these equations can be used, in practice they give much more accuracy when rewritten in a slightly different way (Oger et al. [2007]). Including the density, for example, in the gradient formula:

$$\bigtriangledown A = \frac{1}{\rho}[\bigtriangledown(\rho A) - A \bigtriangledown \rho], \tag{2.20}$$



(a) Gaussian function

(b) Gaussian function derivative

Figure 2.5: Behavior of Gaussian function $W_{Gauss}(r, 1)$and its spatial derivative in one dimension. As already been stated, the function is even and positive definite, while its derivative is odd.

so that 2.19 becomes:

$$\begin{aligned}\nabla A_S(\mathbf{r}_i) &\doteq \frac{1}{\rho_i}\left(\sum_j^N \frac{m_j}{\rho_j}\rho_j A(\mathbf{r}_j)\nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h) - A(\mathbf{r}_i)\sum_j^N \frac{m_j}{\rho_j}\rho_j \nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h)\right)\\ &= \frac{1}{\rho_i}\left(\sum_j^N m_j A(\mathbf{r}_j)\nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h) - \sum_j^N m_j A(\mathbf{r}_i)\nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h)\right)\\ &= \frac{1}{\rho_i}\left(\sum_j^N m_j [A(\mathbf{r}_j)-A(\mathbf{r}_i)]\nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h)\right)\end{aligned} \tag{2.21}$$

and, reformulating eq. 2.18 in similar manner:

$$\nabla \cdot A_S(\mathbf{r}_i) \doteq \frac{1}{\rho_i}\left(\sum_j^N m_j [A(\mathbf{r}_j)-A(\mathbf{r}_i)]\cdot\nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h)\right). \tag{2.22}$$

Moreover, manipulating and rewriting eq. 2.20 in a symmetrized form:

$$\frac{\bigtriangledown A}{\rho} = \bigtriangledown\left(\frac{A}{\rho}\right) + \frac{A}{\rho^2}\bigtriangledown\rho \Rightarrow \bigtriangledown A = \rho\left(\bigtriangledown\left(\frac{A}{\rho}\right) + \frac{A}{\rho^2}\bigtriangledown\rho\right), \tag{2.23}$$

we have for the gradient equation:

$$\begin{aligned}\nabla A_S(\mathbf{r}_i) &\doteq \rho_i \left(\sum_j^N \frac{m_j}{\rho_j} \frac{A(\mathbf{r}_j)}{\rho_j} \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h) + \frac{A(\mathbf{r}_i)}{\rho_i^2} \sum_j^N \frac{m_j}{\rho_j} \rho_j \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h) \right) \\ &= \rho_i \left(\sum_j^N m_j \frac{A(\mathbf{r}_j)}{\rho_j^2} \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h) + \sum_j^N m_j \frac{A(\mathbf{r}_i)}{\rho_i^2} \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h) \right) \\ &= \rho_i \sum_j^N m_j \left[\frac{A(\mathbf{r}_j)}{\rho_j^2} + \frac{A(\mathbf{r}_i)}{\rho_i^2} \right] \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h)\end{aligned} \tag{2.24}$$

2.2 SPH kernel functions

Although the Gaussian kernel is very simple and accurate in interpolating a set of disordered particle, it has the practical disadvantage of being infinite in extension. In this manner all the particles in the system contribute to the summation in eq. 2.11, even those located at a distance such that their actual contribution is very small. In practical use, when this kernel function is implemented in a numerical code, it is assumed that a particle j gives zero contribution when the distance from the point in which the flow variable is calculated (i.e. the particle i), is greater than $3h$ (Gingold and Monaghan [1977]). However, because of an extended support domain, a large number of particles contribute to the calculation leading to an increase of computational effort. To overcome this issue, many authors use kernel functions spatially of finite extension and more compact (go to zero after a shorter finite distance). Among the most popular there are the spherically symmetric cubic spline function proposed by Monaghan and Lattanzio [1985], called $\beta - spline$ and the quintic *Wendland* kernel (Wendland [1995]):

$$W_\beta(r,h) = \frac{\sigma}{h^\nu} \begin{cases} 1 - \frac{3}{2}u^2 + \frac{3}{4}u^3 & 0 \le u \le 1 \\ \frac{1}{4}(2-u)^3 & 1 < u \le 2 \\ 0 & u > 2 \end{cases} \tag{2.25}$$

$$W_{Wend}(r,h) = \frac{\alpha}{h^\nu} \begin{cases} (1 - \frac{u}{2})^4 (2u+1) & 0 \le u \le 2 \\ 0 & u > 2 \end{cases} \tag{2.26}$$

where ν is the number of dimensions and $u = \frac{r}{h} = \frac{|\mathbf{r}-\mathbf{r}'|}{h} = \frac{|\mathbf{r}_i - \mathbf{r}_j|}{h}$ is the relative distanza between two points (particles) at $\mathbf{r}$ and $\mathbf{r}'$ ($\mathbf{r}_i$ and $\mathbf{r}_j$). The constants σ and α are normalization factors and assume the values: $\sigma = 10/7\pi$, $1/\pi$ and $\alpha = 7/4\pi$, $7/8\pi$ in $2-D$ and $3-D$, respectively. Both of the above kernels

have been used for the simulations described in this work. Other kernel functions description can be found in Liu [2002], Liu and Liu [2003], Monaghan [2005].

2.2.1 Kernel and kernel gradient corrections

So far we discussed the properties the kernel must have to reproduce a function in both *integral* (or *kernel*) and *particle* approximations. While in the first approximation case this capability is guaranteed by the conditions imposed in building the kernel (normalization 2.4 plus symmetry 2.6), there's not the same guarantee in the latter one. This problem is called ' particle approximation inconsistency'. In the discrete particle approximation, conditions of normalization and symmetry are satisfied by the following equations:

$$\sum_{j}^{N} W(\mathbf{r}_i - \mathbf{r}_j, h) \triangle \mathbf{r}_j = 1, \tag{2.27}$$

$$\sum_{j}^{N} (\mathbf{r}_i - \mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h) \triangle \mathbf{r}_j = 0. \tag{2.28}$$

Clearly, these identities are not always respected as in the case of boundary-support domain overlapping (as fluid particles near a free surface) which yields a kernel function truncation (this issue occurs in the integral approximation too). Yet another example is that of an irregular distribution of particles that causes unbalanced contributions in the summations (Fig. 2.6). In these eventualities, to restore the consistency, it is possible to make appropriate corrections to the kernel or its gradient. The most common approach is that proposed by Bonet and Lok [1999], that for the kernel, suggest a correction in such a manner that a vector function may be expressed as:

$$f(\mathbf{r}_i) \doteq \frac{\sum_j^N \frac{m_j}{\rho_j} f(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h)}{\sum_j^N \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h)} \tag{2.29}$$

The same authors suggested a correction for the kernel gradient, in order to correctly evaluate the forces in the equation of motion and preserve the angular momentum. This correction is done via a matrix $\mathbf{L}$ such that:

$$\tilde{\nabla}_i W(\mathbf{r}_i - \mathbf{r}_j, h) = \mathbf{L}_i \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{2.30}$$

where the left hand side (LHS) term represents the corrected kernel gradient

and $\mathbf{L}_i$ is given by:

$$\mathbf{L}_i = \left(\sum_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \otimes (\mathbf{r}_i - \mathbf{r}_j) \right)^{-1}. \tag{2.31}$$

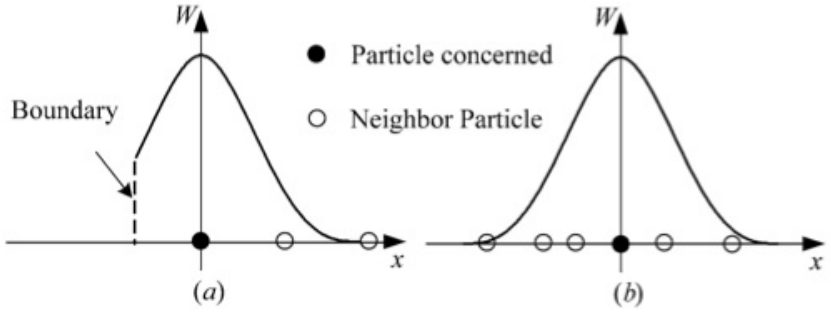


Figure 2.6: Inconsistency of the SPH particle approximation: (a) kernel support domain truncation and (b) irregular particles distribution.

2.3 The Navier-Stokes fluid equations

The governing equations that describe the dynamic of a fluid are represented by the Navier-Stokes (N-S) equations which are the equivalent of Newton's second law applied to fluid motion. Generally, the momentum equation is written alongside the mass and energy conservation. For sake of simplicity[1], here will be considered only the first two laws:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}, \tag{2.32}$$

$$\frac{D\rho}{Dt} + \rho \left(\nabla \cdot \mathbf{v} \right) = 0, \tag{2.33}$$

where ρ is the pressure, $\mathbf{v}$ is the flow velocity, p is the pressure, $\mathbf{T}$ is the viscous stress tensor and $\mathbf{f}$ represents body forces, usually identified with the gravity force. Obviously, the position $\mathbf{r}$ of the particle and its velocity $\mathbf{v}$ are related by the identity:

$$\frac{D\mathbf{r}}{Dt} = \mathbf{v}, \tag{2.34}$$

[1] the temperature effects on the flow are neglected

The writing D/Dt stands for *material derivative* and it is such that:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla. \tag{2.35}$$

If the assumption of *incompressible* flow is taken (ρ=constant), eq. 2.33 reduces to:

$$\nabla \cdot \mathbf{v} = 0. \tag{2.36}$$

With this precondition and assuming constant dynamic viscosity μ, eq. 2.32 can be rewritten as:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}, \tag{2.37}$$

where the operator ∇^2 (the *Laplacian*) is applied to the velocity field in such a way that viscous forces act as a means of diffusion of the momentum, just like the heat diffusion in the heat equation (which also involves the Laplacian operator).

2.4 SPH particle approximation for N-S equations

In this section a complete description of the SPH formulation of the above cited N-S equation, will be given. More specifically, the two different approaches (WCSPH[1], ISPH[2]) used in the development of a numerical code for modeling a Newtonian viscous fluid, will be exposed. Moreover, the fundamental differences between the two methods will be explained, together with the advantages and disadvantages in using the one or the other formulation.

2.4.1 Density equation

As we have seen, applying the SPH formalism to approximate a generic function, the density is described by eq. 2.11. This formulation however, though conserves the mass exactly, generates problems in the vicinity of the fluid edges, decreasing abruptly in a non-natural way. Unfortunately it is not always possible to remedy this, such as on the free surface interface. For this reason, it's preferable to use the continuity equation 2.33 to calculate the density at each point of the fluid

[1] Weakly Compressible Smoothed Particle Hydrodynamics
[2] Incompressible Smoothed Particle Hydrodynamics

domain. So, applying the definition for divergence and after a little algebra, in the particle approximation we have:

$$\frac{D\rho_i}{Dt} \doteq \sum_j^N m_j(\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h). \tag{2.38}$$

An immediate benefit of eq. 2.38 is to calculate the density through the velocity difference between the pairs of interacting particles in the support domain, regardless of where they are. Clearly, if all the particles are stationary or moving with the same speed the density does not change, just as it should be. All these speeches clearly apply to the case in which we are modeling a (weakly) compressible fluid, i.e. in the WCSPH procedure. On the other hand, to simulate an incompressible fluid with the ISPH approach, is sufficient to consider the density $\rho =$ constant throughout the simulation, with no other prescription.

2.4.2 Momentum equation

The momentum conservation law for a continuos medium is given by eqs. 2.32, 2.37 depending on whether it is a compressible or incompressible fluid description (no turbulence model has been implemented). In both cases, a key part for solving the flow dynamic is the calculation of the forces of pressure. Although the calculation of the pressure gradient may be common to both procedures, the calculation of the pressure itself follows two quite different approaches. Again, following the SPH particle approximation 2.24 for gradient, the pressure related part of momentum, equation, divided by density ρ, become:

$$\left(-\frac{1}{\rho}\nabla p\right)_i \doteq -\sum_j^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij}\right) \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{2.39}$$

where p_x and ρ_x, are pressure and density taken at the position of particle with index $x = i, j$. The term Π_{ij} is an artificial viscous term added to manage sharp variations in the flow that may lead to non-physical numerical oscillation around the areas of interest, such as shock front between fluid and rigid walls. This numerical artifice was first suggested in the works of Monaghan and Gingold [1983], Monaghan and Poinracic [1985]. In the particle approximation, the artificial viscosity is given by the sum of a linear and a quadratic term:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c_S \lambda_{ij} + \beta \lambda_{ij}^2}{\rho_{ij}} & \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0 \\ 0 & \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} \geq 0 \end{cases} \tag{2.40}$$

where

$$\lambda_{ij} = \frac{h_{ij}\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2 + \xi^2}, \tag{2.41}$$

$$\bar{\rho}_{ij} = \frac{\rho_i + \rho_j}{2}, \tag{2.42}$$

$$h_{ij} = \frac{h_i + h_j}{2}, \tag{2.43}$$

$$\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j, \mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j. \tag{2.44}$$

The meaning of terms h, $\mathbf{v}$, $\mathbf{r}$ and ρ has already been described, while c_S indicates the speed of sound in the fluid medium, the value of which is obtained from the relation $c_S = \sqrt{\frac{K}{\rho}}$[1]. The constants α and β are free parameters that have to be set depending of the problem to be studied and $\xi = 0.1h$ is added to avoid that the denominator can be zero, when $\mathbf{r}_i = \mathbf{r}_j$. The artificial viscosity to has nothing to do with the physical viscosity which acts at molecular scales, it serves to dissipate the moment (converting mechanical energy into thermal energy), on one hand and to avoid particles interpenetration, keeping also free surface flows numerically stable, on the other. It is worth to notice, in fact, that the artificial viscosity is active only when the particles are approaching each other (eq. 2.40). So, the viscosity associated with the linear term acts as a bulk viscosity, while the quadratic one is somewhat similar to that developed by von Neumann and Richtmyer [1950], widely used in grid methods, which has the purpose of avoiding particles interpenetration when they are moving at high Mach number. The main reason for this issue lies in the fact that the SPH particles do not have a physical dimension, but numerical only.
In general, when not otherwise explicitly stated, both artificial viscosity terms were retained for WCSPH models, while quadratic term only was used for ISPH modelization.

2.4.3 Pressure evaluation

So far, it has been described the SPH formulation of pressure forces, without actually explaining how this fundamental flow variable is obtained. As already mentioned in the introduction, the WCSPH and ISPH approaches follow radically different paths in solving for pressure at any point (particle) spread over the domain.

[1] K = compressibility constant, ρ = fluid medium density

2.4.3.1 Pressure in WCSPH

Though theoretically treated as incompressible, as is common knowledge, the fluids have the property to be weakly compressible media. Since the motion of a fluid (and thus of the particles) is driven by the pressure gradient, it is crucial to solve this quantity accurately. Batchelor [1967] suggested an equation of state (EOS) for water describing the propagation of sound waves in the medium. Monaghan [1994], modified a little this equation lowering the value of the sound speed, to fit with the numerical models. A choice like this becomes obliged to make the time step (using the Courant condition on the sound speed) is not too small during the simulation. Therefore, the EOS which relates pressure and density of the fluid, is:

$$p = B\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right), \tag{2.45}$$

where $\rho_0 = 1000kg \cdot m^{-3}$ is the reference density and $\gamma = 7$. The coefficient $B = c_0^2\rho_0/\gamma$ is the critical parameter which have to be set to properly fit the case of study, because it determines the maximum change of the density. Since for the density relative variation δ we have:

$$\delta = \frac{\rho - \rho_0}{\rho_0} = \frac{v^2}{c_0^2} = M^2, \tag{2.46}$$

where v is fluid bulk velocity and M is the Mach number, the imposition that the relative change in density does not exceed the 1%, allows to assign a suitable value to the speed of sound c_0. Converting the eq. 2.45 to the particles discrete approximation is a rather straightforward task:

$$p_i = B\left(\left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1\right). \tag{2.47}$$

In their work, Molteni et al. [2007] suggested an alternative equation of state to improve consistency and accuracy of the method. Their formulation takes into account the specific internal energy e and reads as:

$$p = c_0^2\left(\rho - \rho_0\right) + \Gamma\left(e - e_0\right)\left(\rho - \rho_0\right), \tag{2.48}$$

where $\Gamma = \left(\partial c^2/\partial e\right)_{e=e_0}$ is a parameter that indicates the increase (decrease) of the sound speed and then of the pressure, according to internal energy and the subscript "0" denotes the values of variables when the fluid is at rest. Eq. 2.48 has proved useful in damping out pressure oscillations in case of fluid-wall violent impact. The advantage of using this formulations for the pressure lies in the easy implementation into a numerical code. On the other hand, this implies the need

to use very small time step and the occurrence of acoustic waves through the medium, which disturb the flow dynamics.

2.4.3.2 Pressure in ISPH

As already mentioned in section 2.4.1, to force a complete incompressibility of the fluid medium, the density should be given a constant value. Under this constraint, the equation 2.33 reduces to the condition:

$$\nabla \cdot \mathbf{v} = 0 \tag{2.49}$$

To keep a divergence-free velocity field, and then modeling an incompressible fluid, Cummins and Rudman [1999] used the *projection method*, first suggested by Chorin [1968]. The method is based on several logical steps. Starting from the computation of a temporary particle position $\mathbf{r}_i^*$ from known initial position and velocity $\mathbf{r}_i^n$, $\mathbf{v}_i^n$:

$$\mathbf{r}_i^* = \mathbf{r}_i^n + \mathbf{v}_i^n \triangle t, \tag{2.50}$$

an intermediate velocity is evaluated, by the momentum equation at position $\mathbf{r}_i^*$, without considering the pressure forces:

$$\mathbf{v}_i^* = \mathbf{v}_i^n + \left(\nu \nabla^2 \mathbf{v}_i^n + \mathbf{f}_i^n\right) \triangle t, \tag{2.51}$$

where $\nu = \frac{\mu}{\rho} = 10^{-6} m^2/s$ is the kinematic viscosity (considered constant). Now, projecting $\mathbf{v}_i^*$ onto a divergence-free velocity field (eq. 2.49) and a curl-free pressure gradient field, it follows:

$$\mathbf{v}_i^* = \mathbf{v}_i^{n+1} + \frac{\triangle t}{\rho} \nabla p_i^{n+1}. \tag{2.52}$$

Taking the divergence of LHS and RHS of the above equation, and remembering eq. 2.49, we obtain the pressure Poisson equation:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p_i^{n+1}\right)_i = \frac{1}{\triangle t} \nabla \cdot \mathbf{v}_i^*, \tag{2.53}$$

which resolved for p_i^{n+1}, allows to update velocity and position to their final values:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^* - \frac{\triangle t}{\rho} \nabla p_i^{n+1}, \tag{2.54}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \left(\frac{\mathbf{v}_i^{n+1} + \mathbf{v}_i^n}{2}\right) \triangle t. \tag{2.55}$$

The whole algorithm presented, has the form of a *predictor-corrector* method with an intermediate velocity field $\mathbf{v}^*$ resulting from the predictor step in eq. 2.51 and the corrector one in eq. 2.54. The key role in this method is represented by the Poisson equation for the pressure. Exploiting the SPH particle approximation of the Laplacian and divergence operators, it becomes:

$$\sum_j^N 2\frac{m_j}{\rho_j^2}\frac{\left(p_i^{n+1}-p_j^{n+1}\right)\mathbf{r}_{ij}\cdot\nabla_i W_{ij}}{|\mathbf{r}_{ij}|^2+\xi^2}=\frac{1}{\triangle t}\sum_j^N\frac{m_j}{\rho_j}\left(\mathbf{v}_j^*-\mathbf{v}_i^*\right)\cdot\nabla_i W_{ij}, \tag{2.56}$$

where, for abbreviation, $W(\mathbf{r}_i-\mathbf{r}_j,h)$ has been replaced with W_{ij}.
Eq. 2.56 leads to solve a linear system of the type $\mathbf{Ax}=\mathbf{b}$, in which:

$$A(i,j)=-2\frac{m_j}{\rho_j^2}\frac{\mathbf{r}_{ij}\cdot\nabla W_{ij}}{|\mathbf{r}_{ij}|^2+\xi^2}, \tag{2.57}$$

$$A(i,i)=-\sum_j^N A(i,j), \tag{2.58}$$

$$x_i=p_i, \tag{2.59}$$

$$b_i=\frac{1}{\triangle t}\sum_j^N\frac{m_j}{\rho_j}\left(\mathbf{v}_j^*-\mathbf{v}_i^*\right)\cdot\nabla W_{ij}. \tag{2.60}$$

If a particle is located on the free surface, it is assigned a pressure equal to zero according the Dirichlet boundary conditions.
A variant of the divergence-free velocity method, is that proposed by Shao and Lo [2003] and previously by Koshizuka and Oka [1996] in the context of the moving particle semi-implicit method (MPS). In this case, the projection method is used to impose an invariance of the density to the incompressible fluid: density is not forced to be constant during the simulation as before. A temporary velocity field is calculated neglecting pressure forces, as in eq. 2.51, and used to update the position to a temporary value:

$$\mathbf{r}_i^*=\mathbf{r}_i^n+\mathbf{v}_i^*\triangle t. \tag{2.61}$$

At this position, $\mathbf{v}_i^*$ is projected onto $\mathbf{v}_i^{n+1}$ and ∇p^{n+1} fields, as in eq. 2.52. The biggest difference in the resulting Poisson equation, stands in the source term, i.e. the RHS equation term:

$$\nabla\cdot\left(\frac{1}{\rho^*}\nabla p_i^{n+1}\right)_i=\frac{\rho_0-\rho^*}{\rho_0\triangle t^2}, \tag{2.62}$$

where ρ_0 and ρ^* are the reference and temporary particle densities, calculated through the summation density equation 2.11. Finally, the velocity can be corrected as:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n - \frac{\triangle t}{\rho^*} \nabla p_i^{n+1}, \tag{2.63}$$

and the position updated trough the eq. 2.55.
The two different approaches have their own drawbacks: in the first (divergence-free velocity) one, density variation or particle clustering, due to the kernel flaw, can occur making the method unstable. This is particularly likely at high Reynolds number. The second one, (density-invariance) is stable, but noisy and inaccurate. A solution, as suggested by Hu and Adams [2007], would be to combine the two methods to overcome all their problems, but the computational cost in solving two Poisson equations is too high. Xu et al. [2009], showed that the divergence-free method, coupled with a technique to artificially move the particles in a suitable manner, can avoid irregular particle spacing and clustering and then carry out accurate results. This was the line of development followed in the present work.
Summarizing, the ISPH approach avoids the use of any state equation, the corresponding algorithm is more accurate but more complex since the pressure must be calculated implicitly. Moreover, the resolution of the Poisson equation requires significant computational costs, compared to the weakly compressible counterpart. However, the noticeable advantage is that the time step can be increased by an order of magnitude.

2.4.4 The viscous forces

In the N-S equations, viscous stresses are modeled according specific assumptions on the fluid behavior which allow to express these type of forces in terms of flow variables, such as velocity and density. Mathematically, the friction forces in a viscous fluid are proportional to the divergence of the *deviatoric* (viscous) stress tensor $\mathbf{T}$ which, in turn, depends on the velocity field tensor gradient. In addition, the fluid is assumed to be isotropic so that $\mathbf{T}$ is a symmetric tensor and vanishes when there is no flow (fluid at rest). With a constant dynamic viscosity coefficient μ, $\mathbf{T}$ has the form:

$$\mathbf{T} = \mu \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T - \frac{2}{3} (\nabla \cdot \mathbf{v}) \mathbf{I} \right), \tag{2.64}$$

where $\mathbf{I}$ is the 3x3 identity matrix. Using the index Einstein notation $\mathbf{T} = \mu \varepsilon^{\alpha\beta}$ and switching to particle approximation, the viscous term $\nabla \cdot \mathbf{T}$ in N-S momentum

equation becomes:

$$\left(\frac{D\mathbf{v}_i^\alpha}{Dt}\right)_{viscous} \doteq \mu \sum_j^N m_j \left(\frac{\varepsilon_i^{\alpha\beta}}{\rho_i^2} + \frac{\varepsilon_j^{\alpha\beta}}{\rho_j^2}\right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\alpha}, \tag{2.65}$$

where:

$$\varepsilon_i^{\alpha\beta} = \sum_j^N \frac{m_j}{\rho_j} \mathbf{v}_j^\beta \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\alpha} + \sum_j^N \frac{m_j}{\rho_j} \mathbf{v}_j^\alpha \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} - \left(\frac{2}{3} \sum_j^N \frac{m_j}{\rho_j} \mathbf{v}_j \cdot \nabla_i W_{ij}\right) \delta^{\alpha\beta}, \tag{2.66}$$

with the symbol $\delta^{\alpha\beta}$ representing the Kronecker delta.
What has just been said, is valid in the case we are dealing with a (weakly) compressible fluid. If on the other hand, the flow we are modeling is truly incompressible, the constant density value in the mass conservation equation (2.33) implies that the spatial derivative of the velocity fields vanishes. This remarkably simplifies the mathematical form of the viscous tensor, which (taking a constant viscosity coefficient) can be rewritten as:

$$\left(\frac{D\mathbf{v}}{Dt}\right)_{viscous} = \frac{\nu}{\rho} \nabla^2 \mathbf{v} = \mu \nabla^2 \mathbf{v}. \tag{2.67}$$

The Laplacian operator has already been encountered in the formulation of the Poisson pressure equation. So, the SPH particle approximation form for the shear stress of an incompressible fluid is similar to the LHS of equation 2.56:

$$\left(\mu \nabla^2 \mathbf{v}\right)_i \doteq 2\mu \sum_j^N \frac{m_j \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{\rho_j \left(|\mathbf{r}_{ij}|^2 + \xi^2\right)} \mathbf{v}_{ij} \tag{2.68}$$

2.4.5 The gravity force

In N-S equation (2.32) for the momentum conservation, gravity is counted as a *body* force. Its physical dimensions are those of a force per unit of volume and acts indiscriminately along the vertical direction of the whole fluid domain. There is no difficulty in implementing this force in a SPH numerical code:

$$\left(\frac{D\mathbf{v}_i}{Dt}\right)_{gravity} = \mathbf{g}_i, \tag{2.69}$$

where $\mathbf{g}$ is equal to $(0, 0, -9.81)$ in $3D$ and $(0, -9.81)$ in $2D$.

2.5 Boundary conditions

The issues of the SPH method related to the kernel truncation near the domain boundaries has already been introduced in sections 2.1.2 and 2.2.1. This is a problem suffered by both the integral and particle approximation, causing the surface integral in the RHS of eq. 2.15 not to be zero in the first approach, and a failure to respect the identities 2.27, 2.28 in the latter one (Fig. 2.7). Strictly speaking, there is a problem of particles deficiency near or on the boundary domain which receive a contribution to the summation of close interactions by the particles inside the boundary only (there are no particles outside). This is particularly evident, for example, along a rigid wall where the 'half' contribution do not yields exact solution because, though the velocity vanishes, some other flow variables, such as the density, do not (Fig. 2.8).

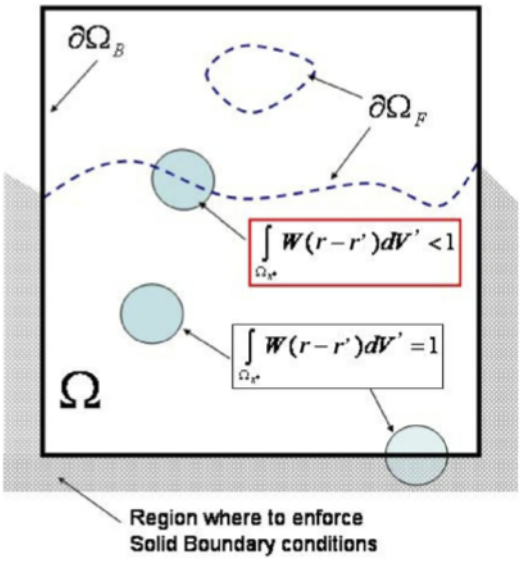


Figure 2.7: Sketch of the different areas in which it may be valid (or not) the normalization condition of the approximation integral. The greek letter Ω indicates the domain volume, while $\partial\Omega$ the surface enclosing it.

Over the years, various strategies have been proposed to fill this lack. Monaghan [1994], used lines of fixed and equally spaced *virtual* particles located right on the boundary to produce repulsive forces, analogous to intermolecular forces, and avoiding the unphysical fluid particles penetration across the boundary. This method was later refined in Monaghan and Kos [1999] to minimize the interspacing effect of the boundary particles on the repulsion of the wall. The repulsion

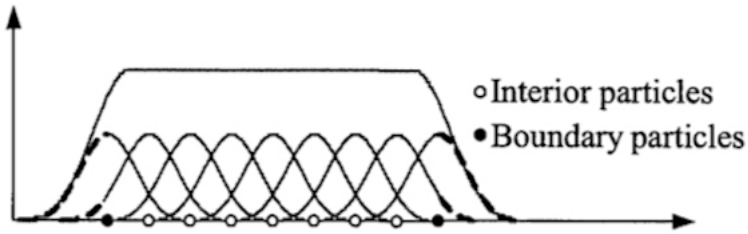


Figure 2.8: SPH kernels interpolation and particle approximation inside and near the boundaries of the domain.

force has the form:

$$\mathbf{f} = \mathbf{n} \cdot R(\triangle n) \cdot P(\Delta t), \tag{2.70}$$

where $\mathbf{n}$ is the normal of the solid boundary, $R(\triangle n)$ and $P(\Delta t)$ are repulsion functions which describe the force experienced by a fluid particle in the directions normal and parallel to the boundary.
Another widely used method, dealing with the boundary conditions treatment, is that of the '*mirror* or *ghost* particles'. This technique, first introduced by Libersky et al. [1993] and later improved by Randles and Libersky [1996] consists in reflecting a symmetrical surface boundary condition, assigning to a fluid particle i, located within a distance κh_i away from the boundary, a mirror particle $i^{'}$ placed symmetrically outside the domain's edge. To satisfy the homogeneous Neumann boundary conditions $\frac{\partial p}{\partial \mathbf{n}} = 0$, these virtual particles have the same density and pressure as the fluid counterparts but opposite normal velocity. So, if the mirror particle $i^{'}$ is the reflection of a fluid particle i, it is assigned the following properties on position r, velocity v and pressure p:

$$r_{i'}^{n} = 2r_{wall}^{n} - r_{i}^{n}, \tag{2.71}$$

$$v_{i'}^{n} = 2v_{wall}^{n} - v_{i}^{n}, \tag{2.72}$$

$$v_{i'}^{t} = \pm v_{i}^{t}, \tag{2.73}$$

$$p_{i'} = p_i + \rho_0 \left(\mathbf{r}_{ii'} \cdot \mathbf{g}\right). \tag{2.74}$$

In the equations above, quotes n and t denote the normal and tangential components, the subscript *wall* indicates the rigid wall, signs + or − are chosen according the *slip* or *no-slip* boundary conditions and in 2.74, the hydrostatic pressure difference between particles i and $i^{'}$ has been added to take into account the effects of the gravity field. The mirror (ghost), repulsive (boundary) and the fluid particles are sketched together in Fig. 2.9. Mirror particles technique has only a weak spot: it becomes difficult to implement in case of complex boundary geometries so that it may be more convenient use other approaches, such as that of '*dummy*' particles. Basically it is a small variation of the ghost particles method: the virtual particles are forced to satisfy the momentum equation, the continuity eqaution, the PPE equation (in case of incompressible fluids), but they do not change their position over the time steps, remaining fixed in preset locations. The dummy particles are arranged on the boundary and in several additional layers inside it, retaining the same pressure and velocity as the corresponding on-boundary particles along the normal direction (Lee [2007]). In the numerical code presented in this work, the repulsive force along with mirror particles technique were implemented although, for the cases presented, only the latter was used.

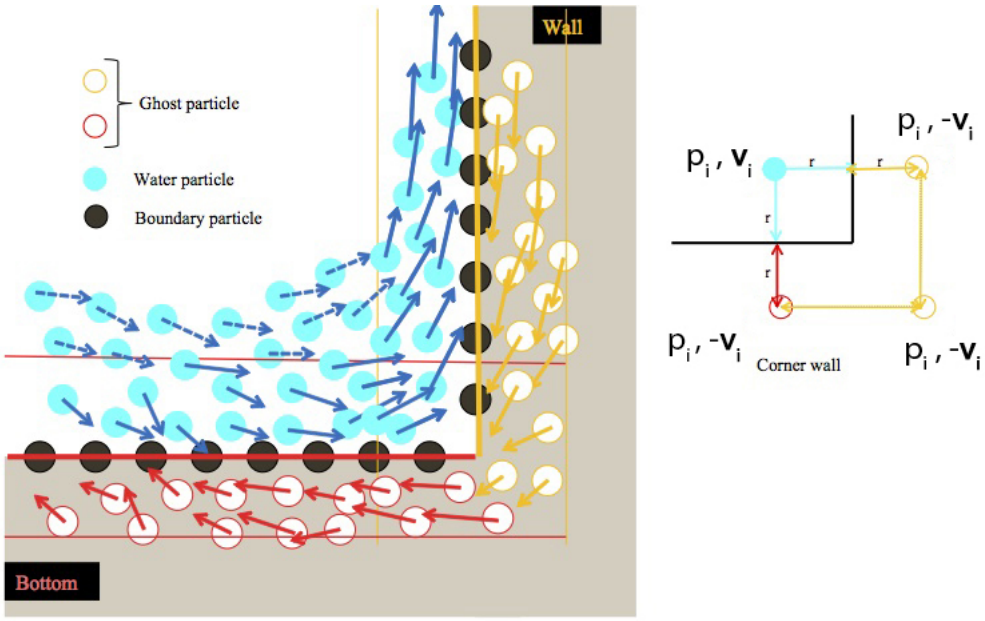


Figure 2.9: Representation of the boundary conditions treatment: repulsive on-boundary particles along with ghost particles, with no-slip conditions (*left*). Zoom of a corner fluid particle reflection, with the ghost technique (*right*).

2.6 Numerical improvements

In this section it will described the measures taken to improve the results of various simulations presented in this work.

2.6.1 Velocity correction

Sometimes, especially for free surface high speed fluid flow or in the case in which there are abrupt solid-liquid interactions that involve excessive fragmentation of the fluid or even a possible fluid particles interpenetration, a small correction on the flow variables can be useful. In his work, Monaghan [1989], proposed a variant in the rate of change of particle position regulated by eq. 2.34, adding a correction term called '*XSPH*'. In the SPH particle formalism this correction is given by:

$$\frac{D\mathbf{r}_i}{Dt} = \mathbf{v}_i + \varepsilon \sum_j^N \frac{m_j \left(\mathbf{v}_j - \mathbf{v}_i\right)}{\bar{\rho}_{ij}} W_{ij}, \tag{2.75}$$

where ε is a parameter varying between 0 (no correction) and 1 (maximum correction). This velocity adjustment has the properties to keep the particles more orderly during the whole simulation.

2.6.2 Linked-list algorithm

In the previous sections it has been emphasized that, each kernel used in the SPH method, has a compact support. This implies that within it fall, at most, a finite number of particles that will then be used in the calculation of the pairs interactions. The particles j within the support domain of a particle i are called *nearest neighboring*. It is clear that, due to Lagrangian nature of the SPH, the nearest neighboring search of a given particle, can vary with time, unlike the grid method where the search of the neighbor cells is determined by their fixed initial position. In a simulation containing N particles, without an efficient search criterion that considers the neighboring only, the number of operations required to calculate the interactions within the domain of a given particle, would be proportional to $(N-1)\cdot N$. The *linked-list* is a search algorithm proposed by Monaghan and Gingold [1983] to reduce considerably the computational costs in the search operations. It consists in temporarily superimpose a mesh, spaced with the kernel support size (Fig. 2.10), adapted to the entire domain of study. Then, once the particles are identified and counted in each cell, the calculation of the 'useful' interaction of a given particle is made considering only the particle's cell and the immediate neighboring ones. Therefore, if the cell spacing is $2h$ the

search is confined only over 3 9 and 27 neighboring cells, in $1-D$, $2-D$and $3-D$ respectively. It has been proved (Monaghan and Lattanzio [1985]) that this procedure reduces the operations cost to $N \cdot log(N)$.

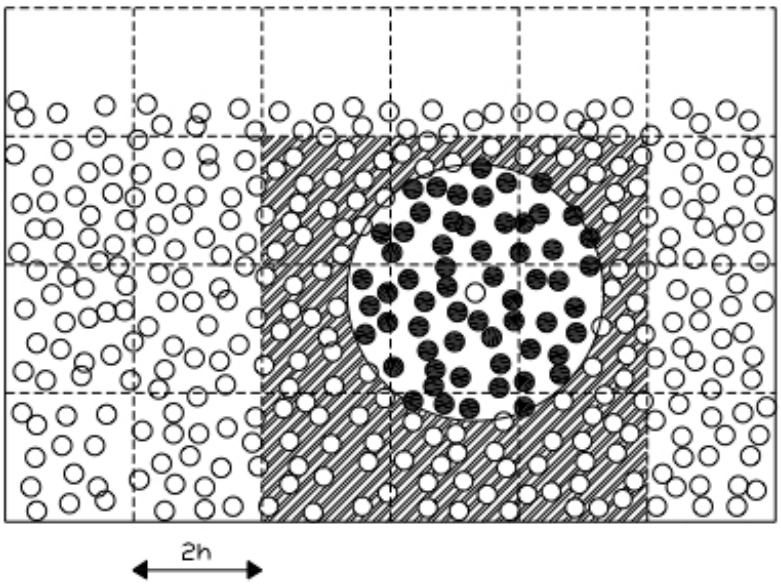
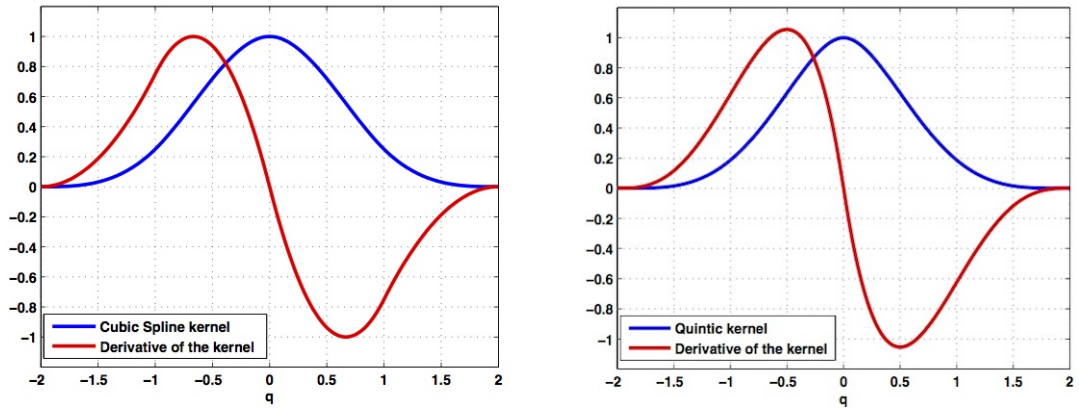


Figure 2.10: Sketch of the temporary $2-D$ mesh overlaid over the problem domain. The cells spacing is $2h$ and if the particle (black dot) is located into the central cell, the neighboring search is confined to 8 adjacent cells only, plus the particle's cell.

2.6.3 Particles shifting (ISPH)

This method was presented in the work of Xu et al. [2009], to improve accuracy and stability in the incompressible flow modeling, when particles clustering occurs. It was showed that in the ISPH method, where a divergence-free velocity field is enforced, particles clustering or density variations could happen, compromising severely the simulation results. The particles clustering phenomenon, as stated by Monaghan [2000], is due to the tensile instability which is related to the behavior of many kernel smoothing functions derivative. From the graphs showing the Wendland and cubic functions first derivative at $r = 0$ (those used in the developed numerical codes), it can be seen that if particles are approaching each other within a certain distance range, the interaction between them decreases rather than increasing (Fig. 2.11). This kernel behavior may lead to large errors in the pressure gradient or Laplacian operator, besides the already mentioned



(a) cubic function (b) Wendland function

Figure 2.11: Behavior of the cubic spline and Wendland functions, along with their spatial derivative in one dimension. The x-axis is in $q = \frac{r}{h}$ units.

particles clumping. Moreover, as was pointed out by Fang and Parriaux [2008], the uneven particles distribution increases the numerical errors. The method consist in moving slightly the particles away from the streamlines, where they tend to accumulate, making the distribution more uniform.

The particle shifting method is divided into the following logical step: once the velocity and position of the particle have been advanced in time in the *corrector* step (eqs. 2.54, 2.55), the same particle is moved slightly from its position by an amount equal to:

$$\delta \mathbf{r}_i = \sigma \left(v_{max} \triangle t \right) \mathbf{R}_i, \tag{2.76}$$

where σ is a parameter to be set, oscillating between $0.01 - 0.1$, $\triangle t$ is the time step, (v_{max} the modulus of the maximum particle velocity and $\mathbf{R}_i$ the *shifting* vector, defined as:

$$\mathbf{R}_i = \sum_j^N \frac{\bar{r}_i^2}{r_{ij}^3} \mathbf{r}_{ij}, \tag{2.77}$$

with N is, as usually the number of neighboring of particle i, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $\bar{r}_i$ is the particle spacing averaged on the neighborhood of i and reads:

$$\bar{r}_i = \frac{1}{N} \sum_j^N r_{ij}. \tag{2.78}$$

The summation 2.77 measures the regularity of the particles distribution around i, in such a way that the closer particles give a greater contribution than

those more distant, just like the kernel interpolation for fluid particles. The final step is to correct (again) the velocity field accordingly by Taylor expansion:

$$\widetilde{v}_i^{\alpha} = v_i^{\alpha} + \frac{\partial v_i}{\partial x_i^{\alpha}} \cdot \delta r_i^{\alpha}, \tag{2.79}$$

where $\widetilde{v}$ is the corrected velocity and the apex α indicates the vector components. The shifting technique is not applied to the surface particles only, to maintain their dynamics as much realistic as possible.

2.6.4 Free-surface particles tracking (ISPH)

The fluid particles that are located on the free surface or in its vicinity, have a number of neighboring lower than those inside the fluid domain, due to a truncation of the support domain by the boundary surface. This particles property can be exploited for their identification during the simulation, since it is a crucial point, in the ISPH approach, to set the pressure value on the liquid surface equal to zero. Calculating the divergence of $\mathbf{r}$, in the particle approximation, we have:

$$(\nabla \cdot \mathbf{r})_i = \sum_j^N \frac{m_j}{\rho_j} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}, \tag{2.80}$$

it has a value of about 2 in $2-D$ and 3 in $3-D$. Clearly, for the particles along the free-surface this value is considerably lower. It is a common criterion to set a value below which a particle is considered to belong to the surface, this is 1.5 for $2-D$ simulations (Lee et al. [2009]) and about 2.3 in $3-D$. Obviously, this criterion is not flawless and it could happen that a surface particle has a value of $\nabla \cdot \mathbf{r}$, higher than that set. However, it was shown that this occurrence does not invalidate the results of the simulation.

2.6.5 Surface tension (WCSPH)

The method, suggested by Becker and Teschner [2007], to model the surface tension in free-surface weakly compressible flows, is based on molecular cohesion forces. The surface tension, in fact, is nothing more than a force of attraction between the molecules of a fluid medium. Hence, adopting this point of view, the authors scaled the attractive forces using the smoothing kernel as a weighting function. The surface tension, experienced by a fluid particle, can be written as:

$$\left(\frac{D\mathbf{v}_i}{Dt}\right)_{surface} = -\frac{\tau}{m_i} \sum_j^N m_j W_{ij} \mathbf{r}_{ij}, \tag{2.81}$$

where τ is a parameter to be set, depending on the problem study.

Chapter 3

Numerical tests and validation

3.1 Introduction

In order to verify the effectiveness and accuracy of the developed SPH codes in reproducing hydrodynamic phenomena, relative to both compressible and incompressible fluids, a series of numerical tests were performed and discussed in this section. The considered cases cover flows in both laminar and turbulent regime, with free (or not) surface and different boundary implementation. A preliminary test to verify the smoothness of the pressure distribution on a rest water tank was performed. As stated before, all the results showed in this work have been obtained using the Wendland and *Cubic Spline* kernels along, each time, with the more suitable numerical improvements showed in section 2.6.

3.2 Pressure distribution on a $2D$ water tank (ISPH, WCSPH)

As a preliminary test, in this section, are reported the results carried out by the *weakly compressible* (WCSPH) and *incompressible* (ISPH) version of the SPH model, regarding the distribution of the hydrostatic pressure of a still water mass filling to the edge a two-dimensional tank. The tank's measurements (and therefore of the water mass) are 1×1 meters. The values of the parameters, used in both models, are shown in table 3.1.

The only substantial difference in the two configurations lies in the fact that in the weakly compressible model are used a greater number of *mirror* particles, to better stabilize the boundary conditions given by the rigid walls. According to Pascal's principle, the distribution of the hydrostatic pressure along the height,

should agree the following analytic law:

$$p = p_0 + \rho g \left(H - y \right), \tag{3.1}$$

with H the height of the water mass, y the coordinate along the water column at which pressure is measured and p_0 the surface pressure, set to zero in the two models. Figures 3.1, 3.2 show the initial configuration for the WCSPH and ISPH model, respectively. In these figures are also emphasized the large number of mirror particles, in one case and the free surface recognition, in the other. Finally, in Figs.3.3,3.4 are shown the results yielded by the above mentioned models (in the same order), of the water column height versus hydrostatic pressure. The figures also show the expected analytic law 3.1, for an easy comparison. It can be seen that both model produce very good results, though the weakly compressible counterpart is a little bit 'noisy', especially at the tank bottom.

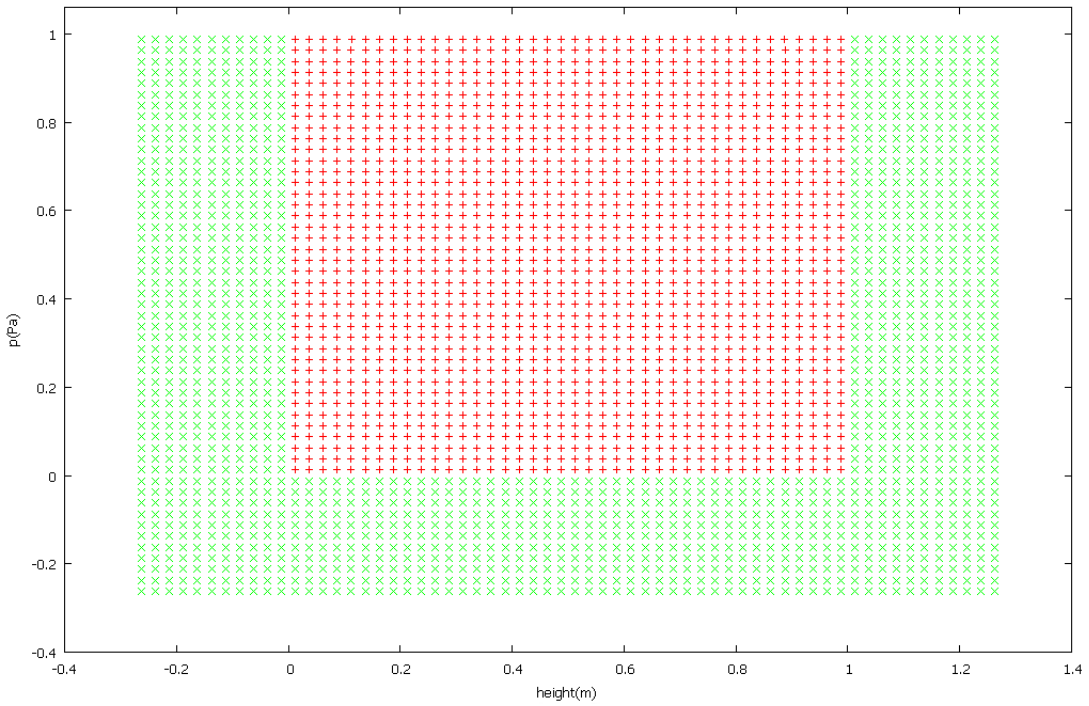


Figure 3.1: Initial configuration for the $2D$ still water tank in the WCSPH model. Green points, representing the rigid walls are made up of *mirror* particles.

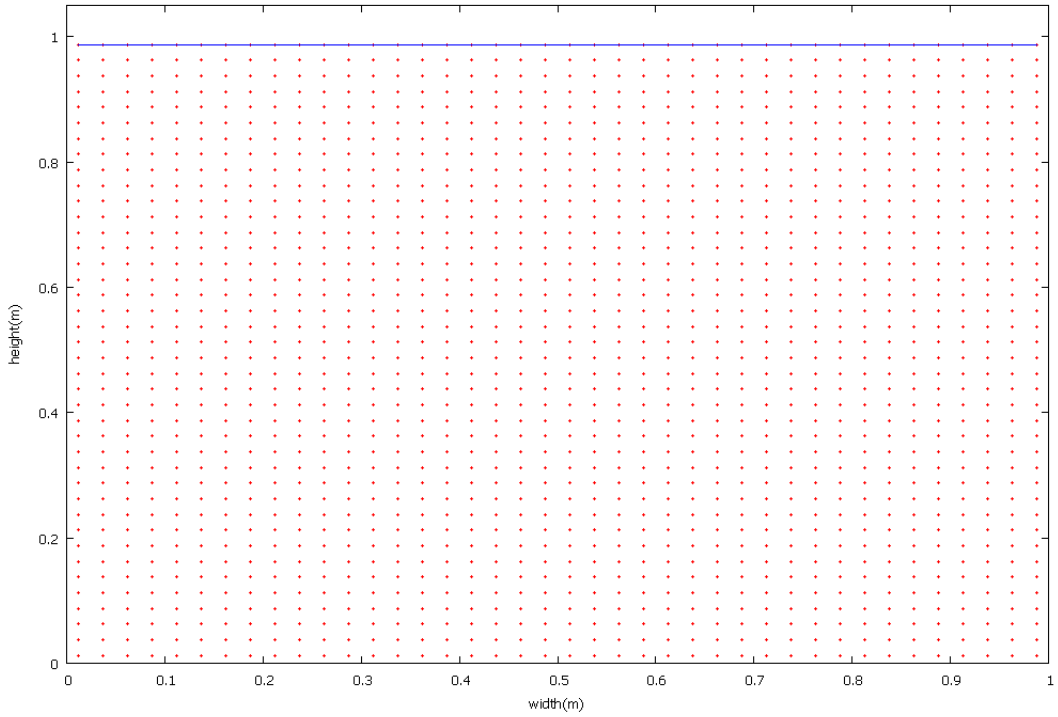


Figure 3.2: Initial configuration for the $2D$ still water tank in the ISPH model. Continuous blue line: free surface recognition.

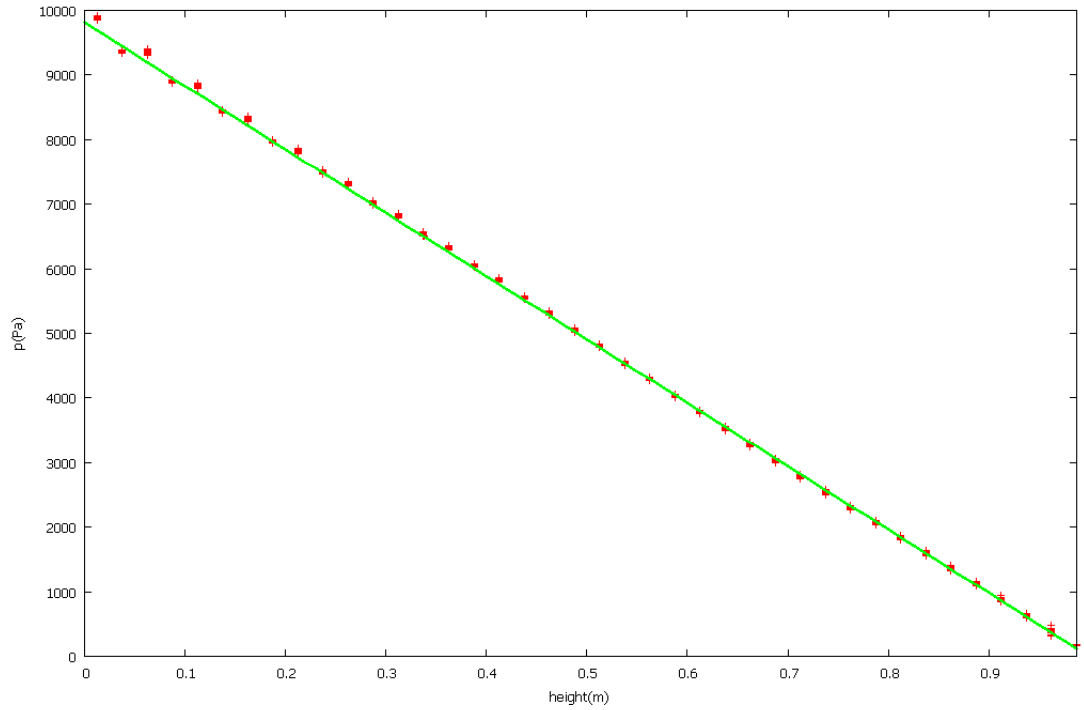


Figure 3.3: Pressure distribution along the water column height in the WCSPH model (red points). Continuous green line: analytic solution according to Pascal's principle.

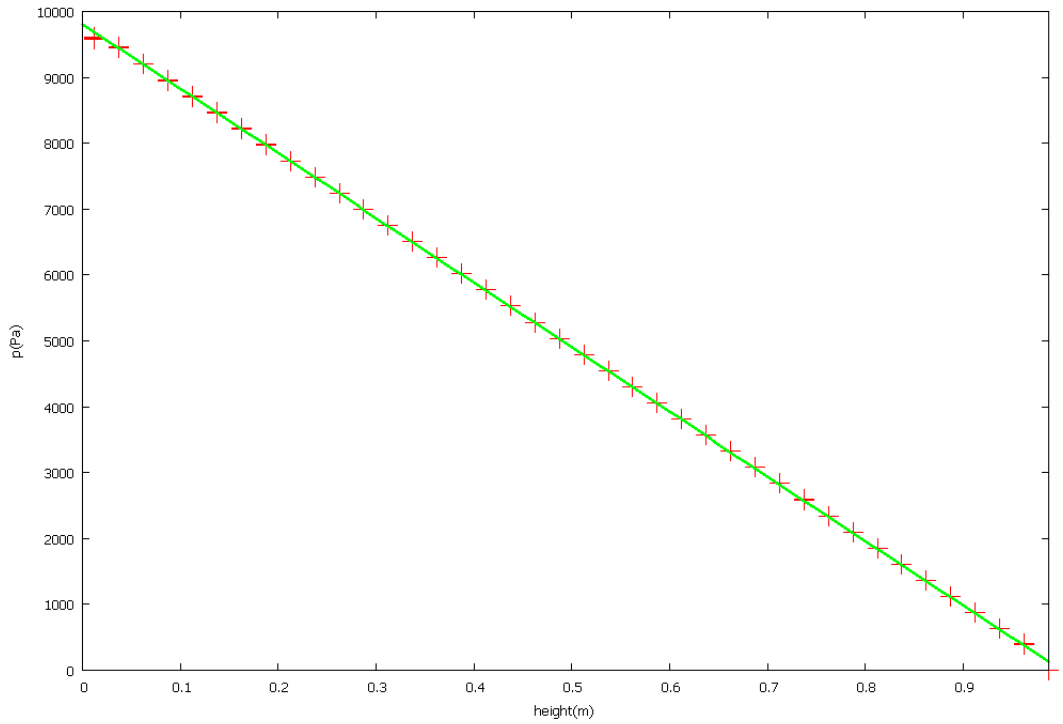


Figure 3.4: Pressure distribution along the water column height in the ISPH model (red points). Continuous green line: analytic solution according to Pascal's principle.

3.3 Surface tension effect on a $2D$ square water drop (WCSPH)

The following test is performed to demonstrate the effects of the surface tension on a fluid. Although the role of this force is not decisive on the overall fluid motion, it proves to be important on the dynamics of the free surface flows. As an example, an experiment was carried out to analyze the evolution of a two-dimensional '*square*' water drop measuring (a, a), in the absence of gravity. The parameters used for the model are in Table 3.2. Fig.3.5 shows the initial conditions of the fluid mass. After some time, under the effect of the internal pressure and the surface tension that exerts a net effect on the more external particles, the results are as shown in Fig.3.6.

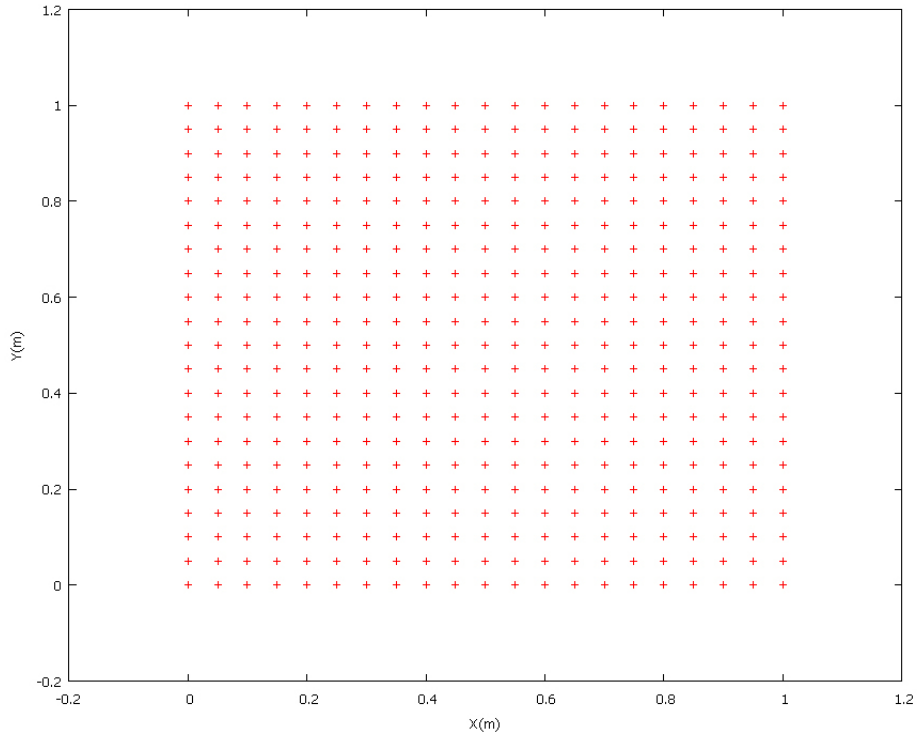


Figure 3.5: Initial water particles (red points) distribution on the $X - Y$ plane.

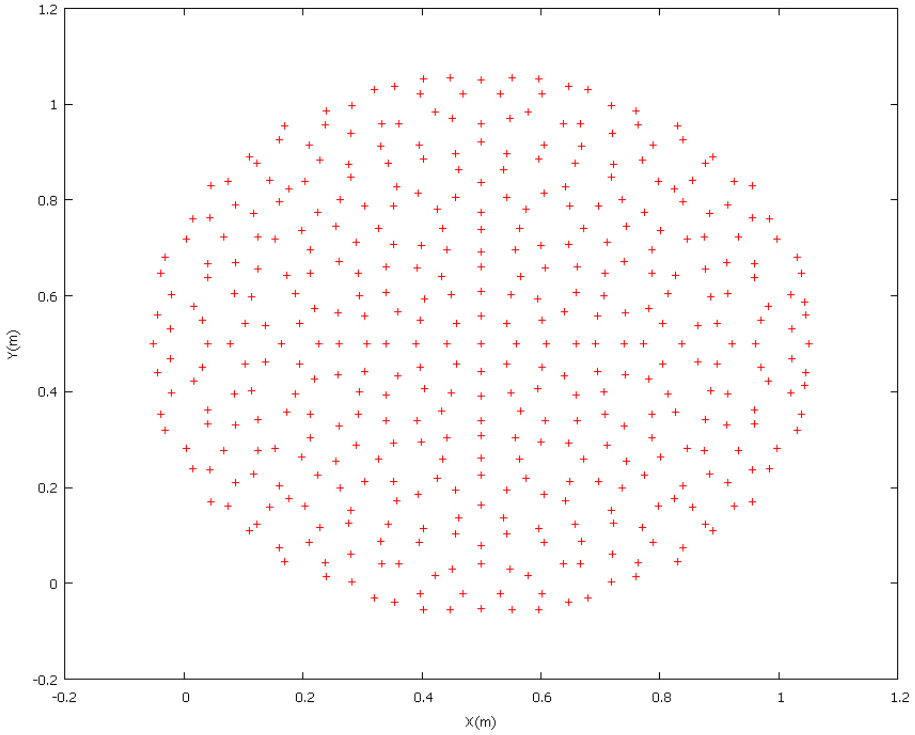


Figure 3.6: Configuration of the water particles (red points) after some time steps, in a state in which the inward surface tension and the outward pressure force are almost balanced

3.4 Laminar flow test cases

In this section, the results produced by simulations of standard cases, such as the lid cavity (also called shear cavity) and the parallel plate flow are showed. In the first case, testing with both the weakly compressible and incompressible version of the SPH were performed, in order to show the consistency in the results. The three-dimensional simulations were produced by integrating the ISPH method in the PANORMUS (http://www.panormus3d.org/developers/) software package, an open source numerical code for the modeling of incompressible fluids, through the of Eulerian grids finite volume technique (FVM).

3.4.1 $3D$ Flow between two parallel planes (ISPH)

The first case presented is the flow in a regular channel having a relatively large cross-section aspect ratio. The problem domain length is $0.01m$ in the streamwise direction (coincident with the x_1 axis), $1m$ and $0.2m$ along x_2 and x_3 axes respectively. The initial particle distribution was very regular, with a particle spacing of $5 \cdot 10^{-3}m$, resulting in a total amount of 16000 effective particles. A Wendland kernel function was used.
Periodic boundary conditions (PBC) have been imposed in the streamwise direction, while the no-slip conditions has been set on the solid walls, using the mirror particles method. A constant mass force (per mass unit) of $0.001m/s^2$ has been imposed in the stream wise direction to drive the flow, while the kinematic viscosity has been set to $10^{-6}m^2/s$. In addition, the particle shifting algorithm has been used to shift the fluid particles at the end of each time step.
The obtained streamwise velocity profile in the central x_3-axis is shown in Fig.3.7, using eq. 2.29 to interpolate the particles velocities on the selected axis. Due to the considered aspect ratio of the channel, the effect of the lateral walls (the ones having a length of $0.2m$) on the profile is negligible. Thus, the analytic solution for the laminar flow between two parallel planes, which is plotted in the figure too, can be used for comparison. As it can be seen, the agreement of numerical results with the analytical profile is very good.

3.4.2 $3D$ Lid cavity (ISPH)

The second considered test case is the classical *lid-driven cavity* problem, where the flow is driven by one of the box walls, sliding on its own plane. The cavity size, the fluid properties and the boundary conditions have been chosen as to reproduce one of the cases analyzed by Albensoeder and Kuhlmann [2005], who used an highly accurate Chebyshev-collocation method to obtain reliable results and to make them available for CFD code validations. Specifically, the domain

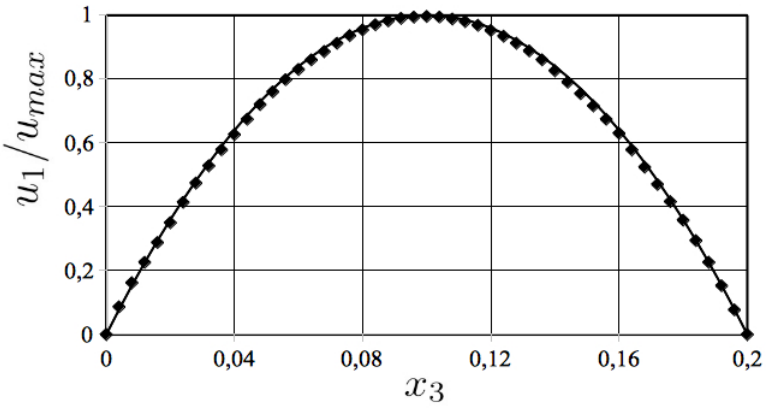


Figure 3.7: Streamwise velocity component non-dimensionalized with the maximum velocity along the central x_3-axis. Continuous line: analytic solution; diamonds: SPH simulation results.

has a cubic shape (Fig. 3.8), with side d equal to $1m$, while the fluid kinematic viscosity has been set to $10^{-4}m^2/s$ and the lid velocity u_s (in the direction x_1) to $10^{-2}m/s$, resulting in a Reynolds number $Re = u_s d/\nu$ equal to 100.

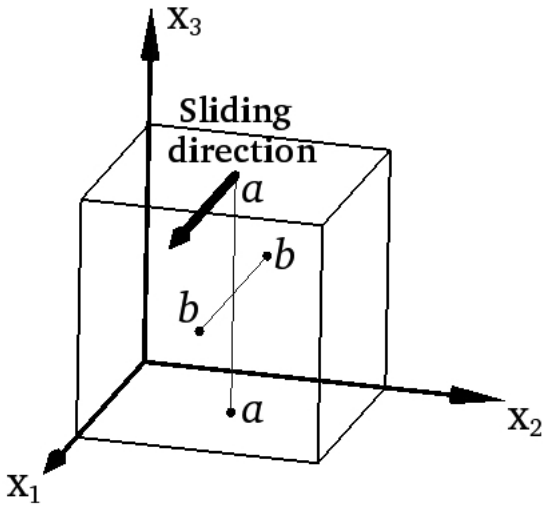


Figure 3.8: Sketch of the $3D$ lid-driven cavity.

The Wendland kernel function width has been set to $d/25 = 0.04m$, while the starting particle distribution has been assigned homogeneously to get a total particles number $N = 125000$. Again, the no-slip boundary conditions via mirror

particles has been imposed at the solid walls and the particle shifting algorithm has been used to avoid the particle clustering along the streamlines.
The profile of the velocity component u_1 in the direction of the sliding wall along the centerline normal to the same plane (axis $a - a$ in Fig. 3.8) is plotted in Fig. 3.9.*a*, together with the results obtained by Albensoeder and Kuhlmann [2005]. In Fig. 3.9.*b* the velocity component, in the direction normal to the sliding plane, is plotted along the central axis parallel to the sliding direction (axis $b - b$ in Fig. 3.8). All the velocities in the profile are made non-dimensional with the lid velocity u_s. In both cases an extremely good agreement is obtained with the reference results, showing the ability of the SPH code to solve hydrodynamically complex flows developing over moving boundaries.

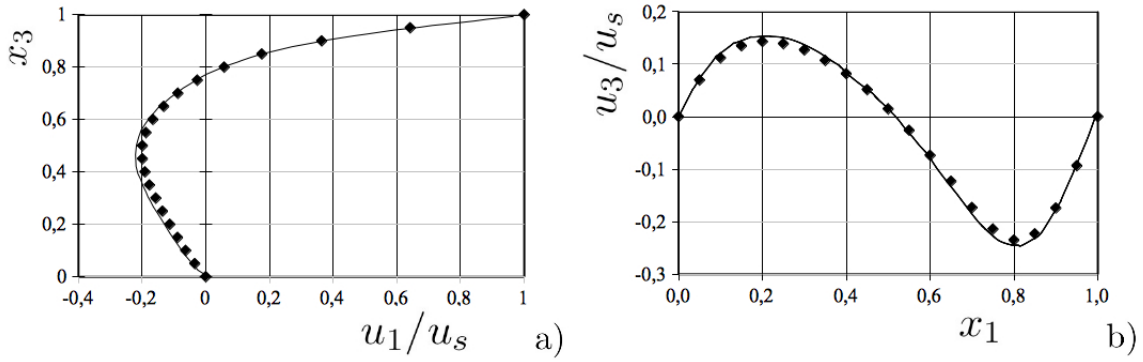


Figure 3.9: Non-dimensional velocity profiles along the cavity axes: a) u_1/u_s profile along the $a - a$ axis; b) u_3/u_s profile along the $b - b$ axis. Continuos lines: reference data from Albensoeder and Kuhlmann [2005]; diamonds: results obtained from the SPH code.

3.4.3 $2D$ Lid cavity (WCSPH)

The lid cavity test was also performed in a pure 2D case, using the weakly compressible variant of SPH. In this case, the considered system was a square box of side $d = 0.001m$ with the upper side moving with a velocity $v = 10^{-3}m/s$. Being the kinematic viscosity $\nu = 10^{-6}m^2/s$, the resulting Reynolds number was $Re = 1$. The initial particle interspacing was set to $2.5 \cdot 10^{-5}m$, in order to have a total fluid particle number $N = 1600$ (Fig.3.10). To solve the physical interactions a Cubic Spline interpolating kernel with a smoothing length equal to the initial particles spacing was exploited. The implemented boundary conditions was those of the repulsive force (see 2.5) exerted by the line of fixed particles that made up the rigid walls. Specifically, to implement the repulsive interac-

tion between fluid and walls, the following simpler formula was used (Monaghan [1994]):

$$F_{ij}^{rep} = \begin{cases} D\left[\left(\frac{r_0}{r_{ij}}\right)^{n_1} - \left(\frac{r_0}{r_{ij}}\right)^{n_2}\right]\frac{\mathbf{r}_{ij}}{r_{ij}^2} & \left(\frac{r_0}{r_{ij}}\right) \leq 1 \\ 0 & \left(\frac{r_0}{r_{ij}}\right) > 1 \end{cases} \tag{3.2}$$

where n_1 and n_2 are set to 12 and 4 respectively, r_0 is the initial particles separation and $D = 0.01$ is a problem depending parameter related to the square of the largest fluid velocity during the simulation. After a few thousand time step, the system reached a steady state configuration. Figs. 3.11 and 3.12 show respectively the field of modules and vectors of the velocity variable at this stage. The obtained results are consistent with those produced with the incompressible SPH counterpart.

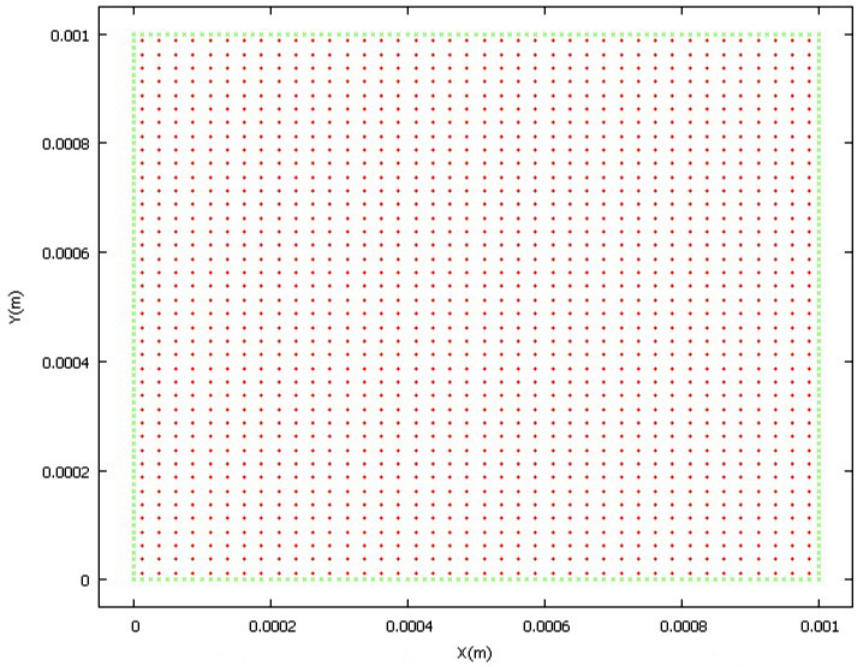


Figure 3.10: Initial particles configuration for 2D weakly compressible SPH simulation of the shear cavity case. Fluid particles are plotted with red dots, while fixed repulsive ones are in green.

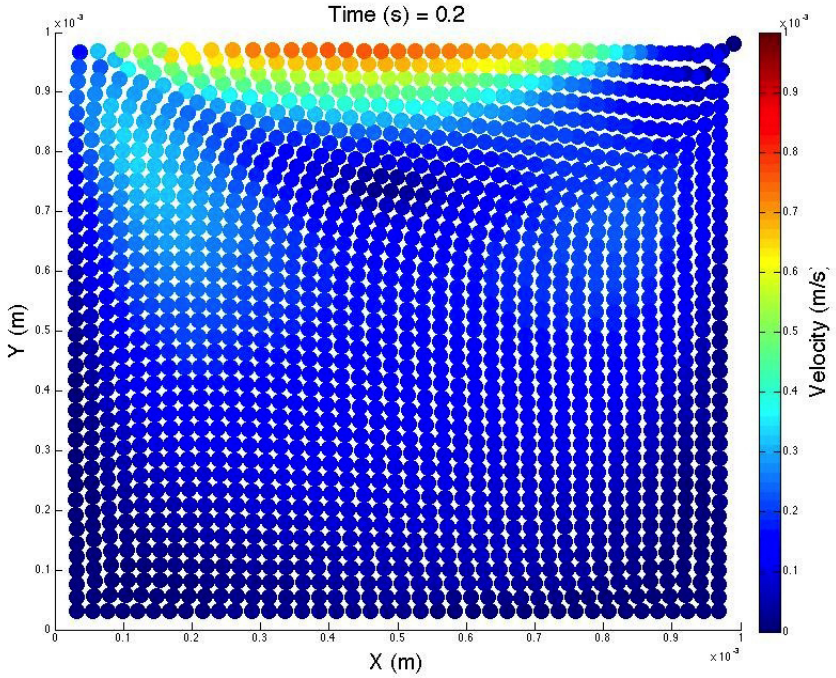


Figure 3.11: Velocity module distribution over the computational domain after $\triangle t = 0.2s$ of simulation, a time interval in which the system has reached a steady state configuration.

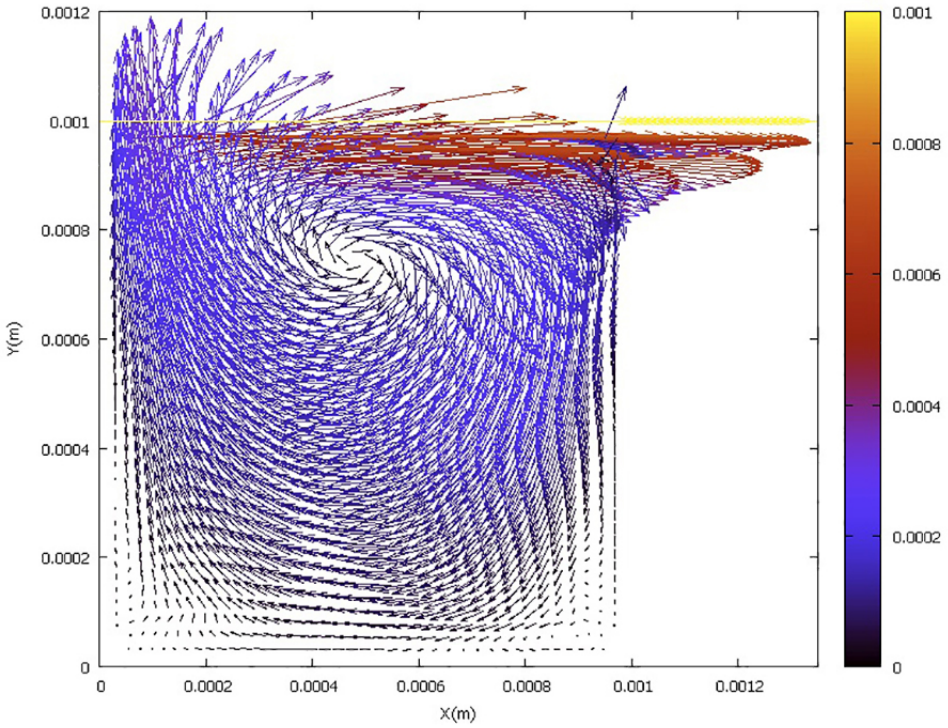


Figure 3.12: Velocity vector field over the computational domain after $\triangle t = 0.2s$ of simulation, a time interval in which the system has reached a steady state configuration.

3.5 $2D$ dam-break

3.5.1 Wall bounded dam-break

The ISPH code developed has been also exploited to analyze the dynamics of, an initially at rest, 2D collapsing water column in a thank under the influence of gravity. This test has also proved useful for the opportunity to compare the results with experimental ones available. The experiment was originally performed by Martin and Moyce [1952] and, more recently, by Koshizuka and Oka [1996] (see details). The whole simulation was carried out choosing the parameter listed in Table 3.3. It was not considered the third dimension (width) since it does not affect the dynamics of the phenomenon. Initial setup of the system is showed in Fig.3.13: the water column, represented by a set of equally spaced points (particles) whose basis and height are respectively a and $2a$, starts its motion from the left vertical wall, spreads over the horizontal plane, then hits the right vertical wall located at position $4a$, where after a short climb in which it shows some fragmentation phenomena, starts moving in the opposite direction. The boundaries have been implemented by means of mirror particles with *free-slip* conditions (Randles and Libersky [1996], Morris et al. [1997]). In Figs. ranging from 3.14 to 3.23 is possible to observe the shapes of the fluid mass in the time interval $0 \div 1$ *secs*. They are clearly in good agreement with the experimental ones seen in shots taken by Koshizuka and Oka [1996] during the same time interval.

3.5.2 Unbounded dam-break

It was also made a slight change to the system configuration eliminating the right vertical wall, in order to let the flow spread over undisturbed upon the longitudinal x direction and make a full comparison with experimental data provided by Martin and Moyce [1952]. The particles shifting technique (Xu et al. [2009]) has been applied together with XSPH method during the whole simulation sessions, keeping a smooth particles distribution. As usually, the left and bottom wall BCs have been implemented with the mirror particles technique (see Table 3.3). Experimental and simulated data are illustrated together in Fig.3.24 and Fig.3.25 which show respectively the position x covered by the wavefront on the horizontal wall and the height z of fluid column on the left side vertical wall, against time. The total simulation time interval was about $0 \div 0.9 secs$. As it can be seen, the obtained results are very good, even better than those reported in previous works.

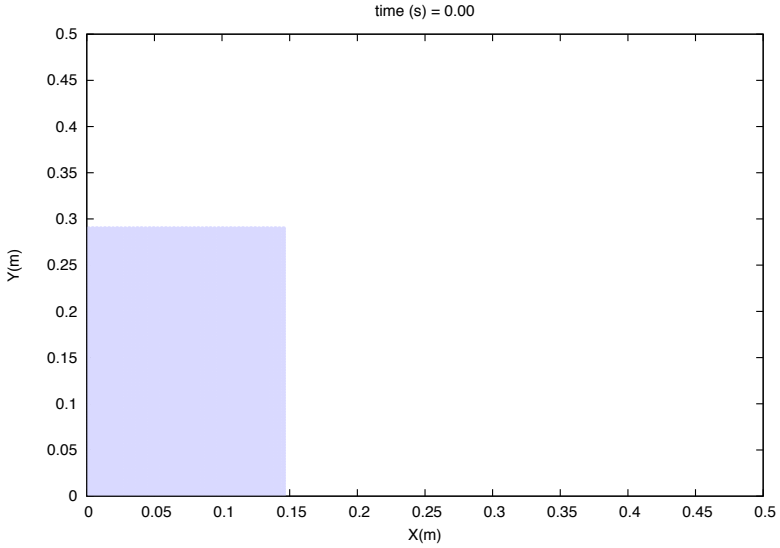


Figure 3.13: Initial setup for $2D$ dam-break experiment. The fluid column is 0.146 m wide and 0.292 m high.

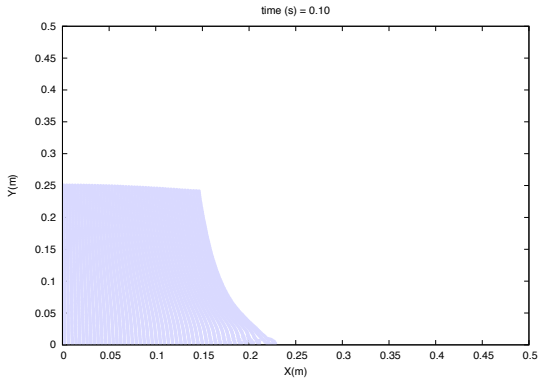


Figure 3.14: t = 0.1 s

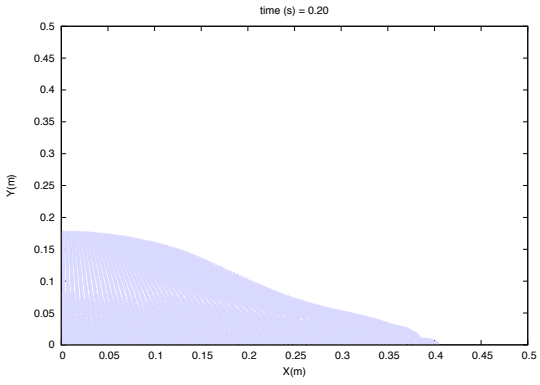


Figure 3.15: t = 0.2 s

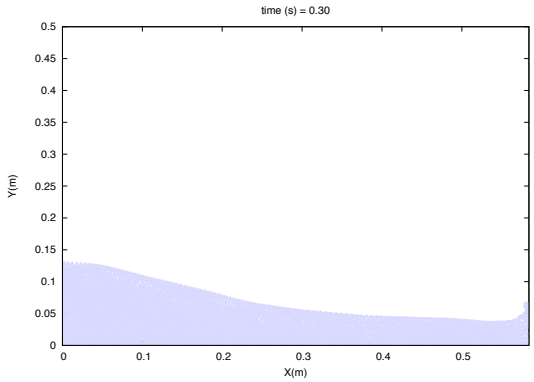


Figure 3.16: t = 0.3 s

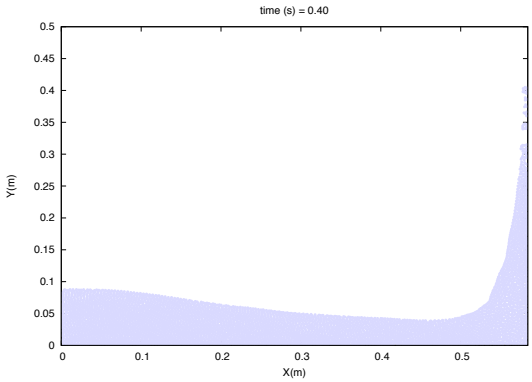


Figure 3.17: t = 0.4 s

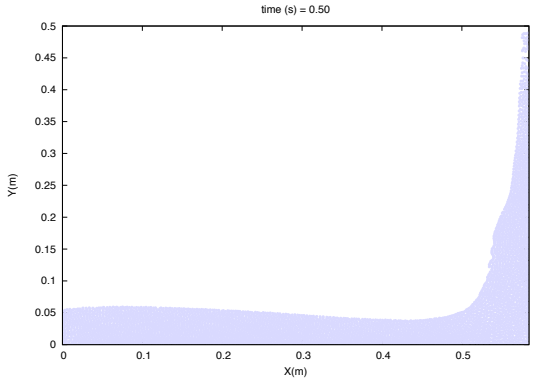


Figure 3.18: t = 0.5 s

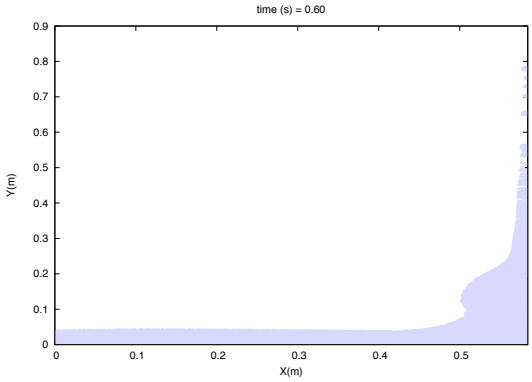


Figure 3.19: t = 0.6 s

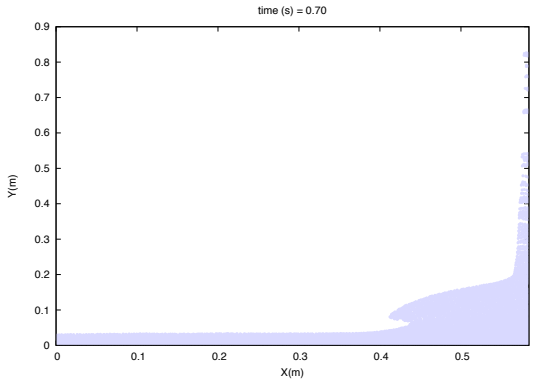


Figure 3.20: t = 0.7 s

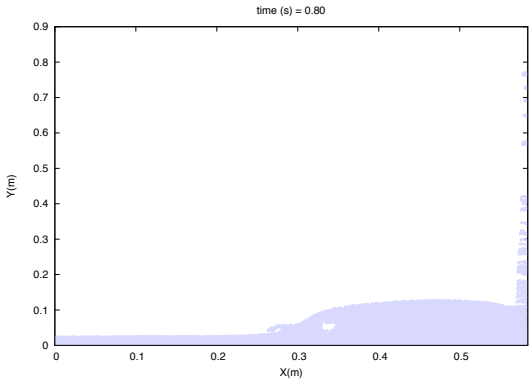


Figure 3.21: t = 0.8 s

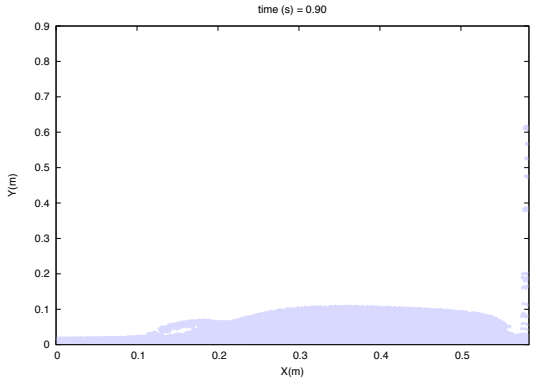


Figure 3.22: t = 0.9 s

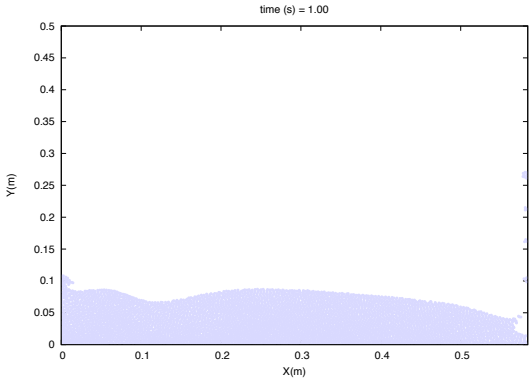


Figure 3.23: t = 1.0 s

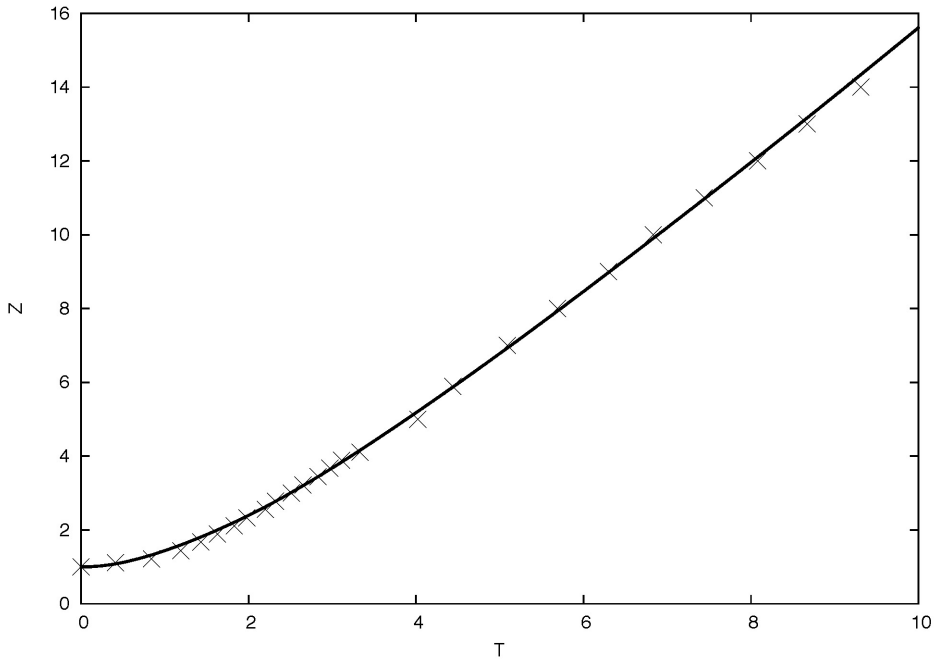


Figure 3.24: Fluid surge front position Z against the time evolution T in dimensionless units $Z = \frac{x}{a}$ and $T = t\sqrt{\frac{2g}{a}}$. Comparison between simulated (solid line) and experimental data (points) by Martin and Moyce [1952].

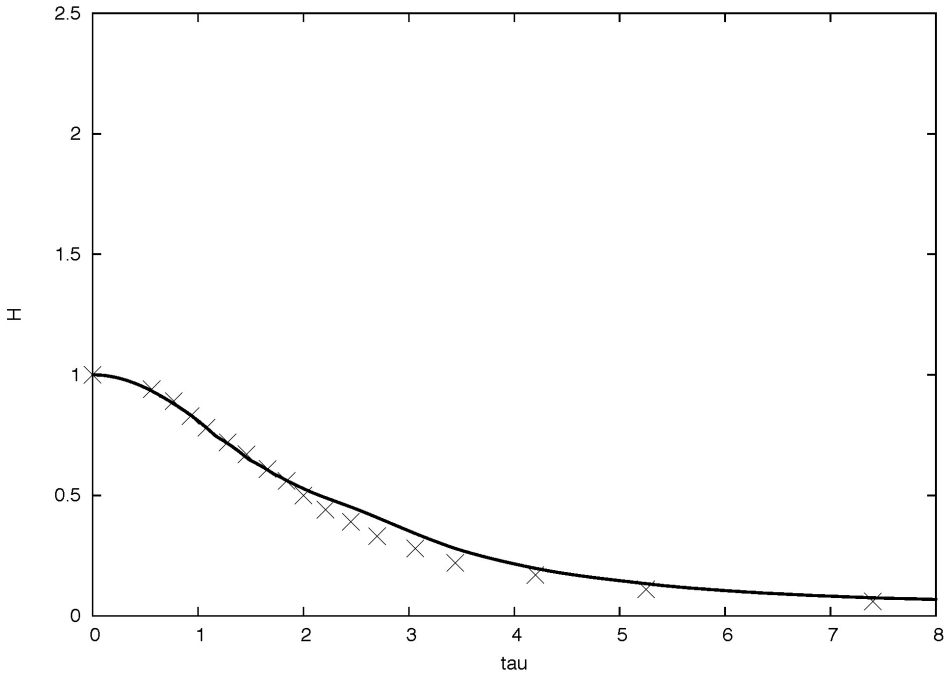


Figure 3.25: Height H of residual fluid column on the right side wall against time evolution τ in dimensionless units $H = \frac{y}{2a}$ and $\tau = t\sqrt{\frac{g}{a}}$. Solid line=simulated data; points=experimental (Martin and Moyce [1952]).

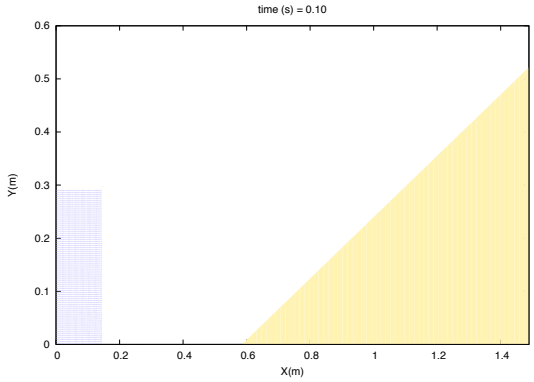


Figure 3.26: t = 0.1 s

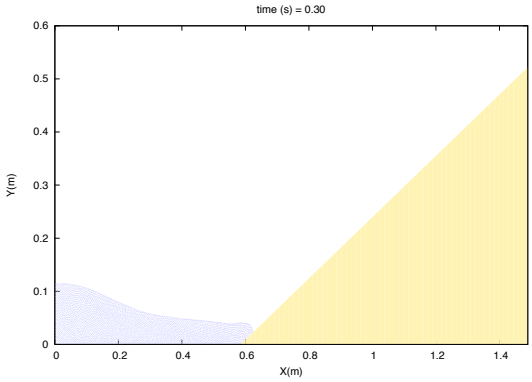


Figure 3.27: t = 0.3 s

3.5.3 $2D$ dam-break with a landslide

To further test the ISPH code, a third simulation sessions was carried out. This numerical experiment has the same initial configuration of the dam-break case, except for the lower particles spatial resolution and the absence of a right vertical wall (see Table 3.4 for the relevant parameters). In place of the aforementioned right boundary condition, a plane tilted at an angle of 30°, has been implemented to simulate a landslide. The fluid mass, of basis a and height $2a$, follows the same dynamycs of the first example, but once reached the horizontal position $4a$, rises along an inclined plane rather than bumping against a vertical rigid wall. The whole system is, again, bounded by means of mirror particles with *free-sleep* condition and the particles shifting and XSPH techniques are used. Figures 3.26 to 3.35 show the system's temporal evolution in the time interval ranging from 0.1 to 1.75 seconds taken with a time step of $0.2s$ (except the last plot). Unfortunately, this case study has neither analytical nor experimental counterpart to compare with.

3.6 $3D$ unbounded flow simulation

3.6.1 $3D$ collapsing water cylinder

Another session of simulation covered the case of an axially-simmetric $3D$ water cylinder collapsing under its own weight (see Table 3.5 for the relevant parameters setting). The fluid mass of radius a and height $2a$ is confined to spread over the horizontal plane. As in the previous case, the continuum is represented by an

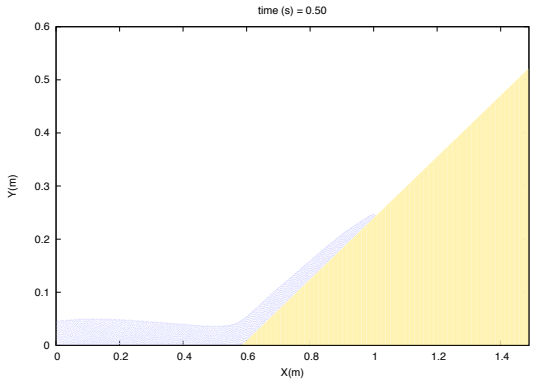


Figure 3.28: t = 0.5 s

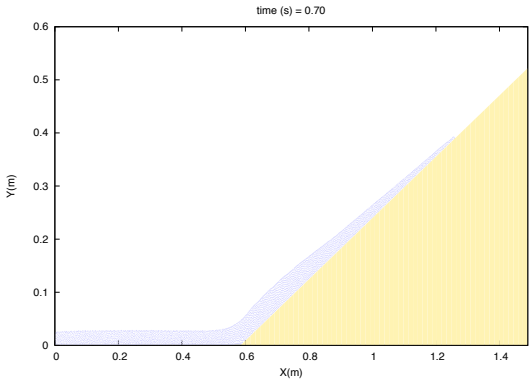


Figure 3.29: t = 0.7 s

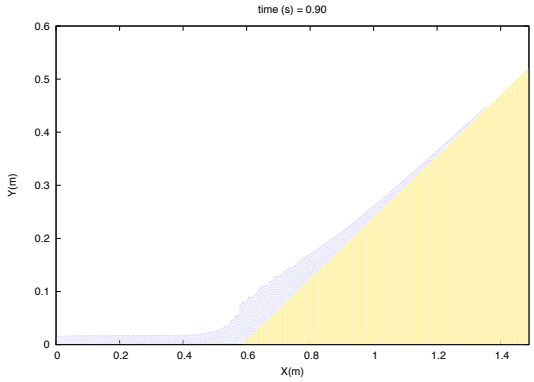


Figure 3.30: t = 0.9 s

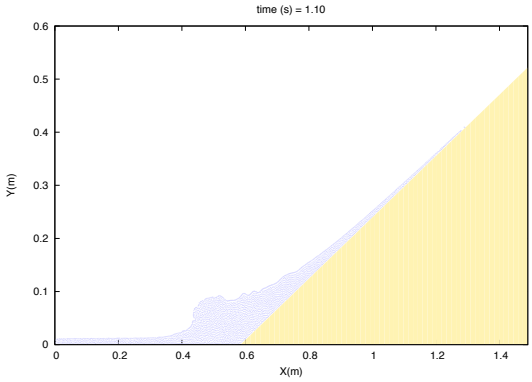


Figure 3.31: t = 1.1 s

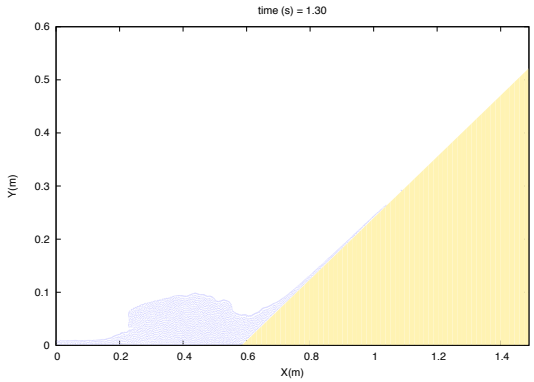


Figure 3.32: t = 1.3 s

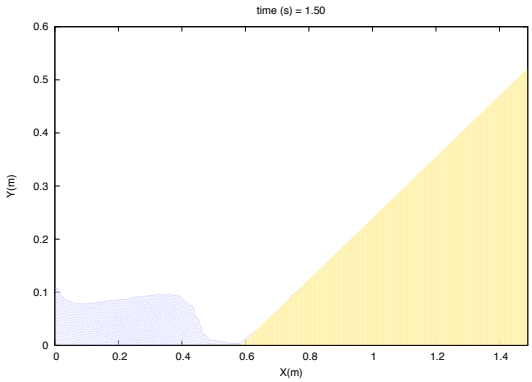


Figure 3.33: t = 1.5 s

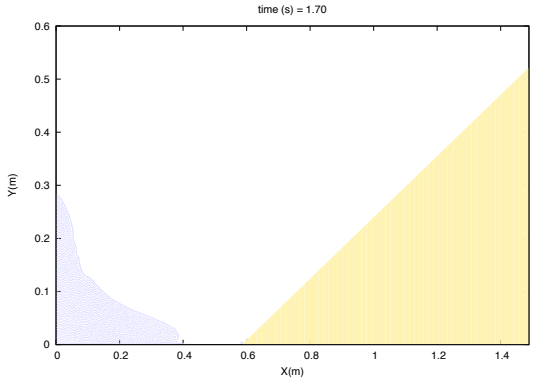


Figure 3.34: t = 1.7 s

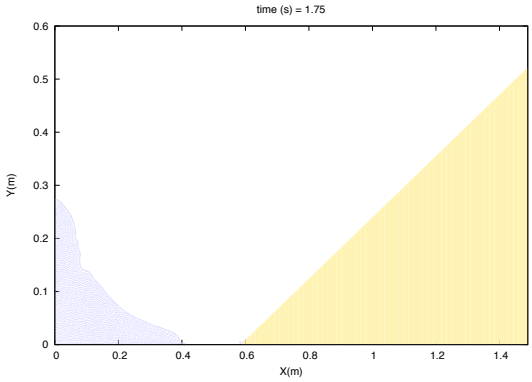


Figure 3.35: t = 1.75 s

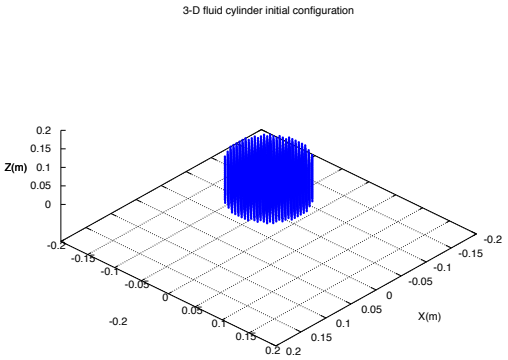


Figure 3.36: Initial setup for $3D$ cylinder collapse experiment. The fluid mass radius and height are respectively 0.06 and 0.12 m.

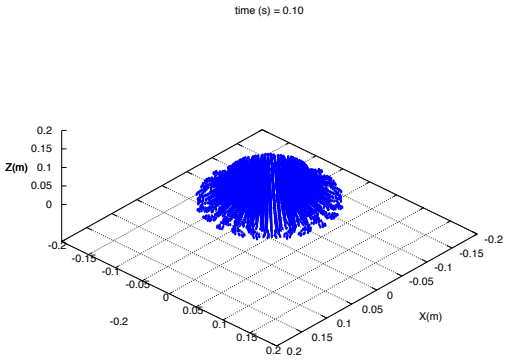


Figure 3.37: Shape of the collapsing fluid cylinder taken at $t = 0.1s$.

initial set of of equally spaced particles along all three axes and the floor is implemented through the mirror particles technique with *free-slip* condition on the boundary interface (see Fig.3.36). The particles shifting and XSPH techniques have been applied together during the whole simulation. We have the following scenario: while the fluid collapses vertically thereby reducing its height, it spreads symmetrically in time along the horizontal plane reaching greater radial distances. The state of the system at time step $t = 0.1s$ is shown in Fig.3.37. In this case too, there was the opportunity of comparing the result obtained with the experimental data from the work of Martin and Moyce [1952]. Since in this case there are no vertical walls, in Fig.3.38 is shown a single graph that relates the time evolution of the wavefront position along radial direction. As in the previous dam-break experiment, the obtained results fit very well to experimental ones.

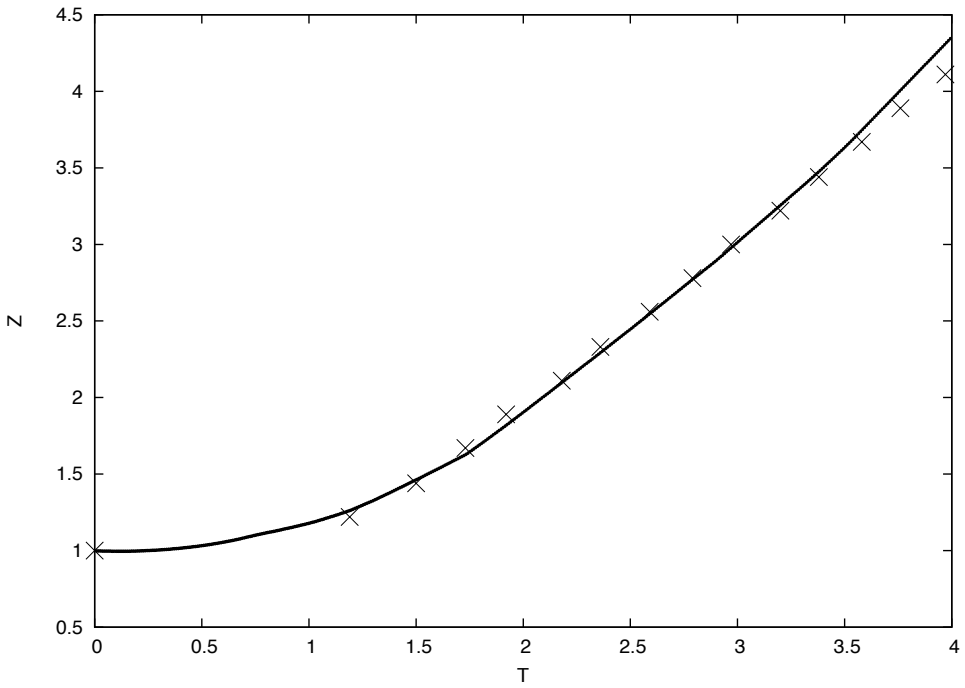


Figure 3.38: Fluid surge front radial position Z against the time evolution T in dimensionless units $Z = \frac{r}{a}$ and $T = t\sqrt{\frac{2g}{a}}$. Comparison between simulated (solid line) and experimental data (points) by Martin and Moyce [1952].

Table 3.1: $2D$ still water tank parameters

parameter	units	value
$heigth{=}width$	(m)	1.0
$\triangle x = \triangle y$	(m)	0.025
$kernel$	(m^{-2})	$Cubic$
h	(m)	$1.3\triangle x$
N	(numb. of particles)	1600

Table 3.2: $2D$ '*square drop*' parameters

parameter	units	value
a	(m)	1.0
$\triangle x = \triangle y$	(m)	0.05
$kernel$	(m^{-2})	$Cubic$
h	(m)	$\triangle x$
N	(numb. of particles)	441
$viscosity$	$(\frac{kg}{m \cdot s})$	0.001

Table 3.3: $2D$ dam-break parameters

parameter	units	value
a	(m)	0.146
$\triangle x = \triangle y$	(m)	0.003
$kernel$	(m^{-2})	$Wendland$
h	(m)	$\triangle x$
N	(numb. of particles)	4753
$viscosity$	$(\frac{kg}{m \cdot s})$	0.001

Table 3.4: $2D$ dam-break with landslide parameters

parameter	units	value
a	(m)	0.146
$\triangle x = \triangle y$	(m)	0.0037
$kernel$	(m^{-2})	$Wendland$
h	(m)	$1.2\triangle x$
N	(numb. of particles)	3081
$viscosity$	$(\frac{kg}{m \cdot s})$	0.001

Table 3.5: $3D$ cylinder collapse parameter settings

parameter	units	value
a	(m)	0.06
$\triangle x = \triangle y = \triangle z$	(m)	0.006
$kernel$	(m^{-3})	$Wendland$
h	(m)	$1.2\triangle x$
N	(numb. of particles)	6320
$viscosity$	$(\frac{kg}{m \cdot s})$	0.001

Chapter 4

Simple absorbing layer conditions for shallow wave simulations with WCSPH

4.1 Introduction

The problem of non-reflecting boundary conditions is an old and important subject of the study of wave propagation in limited domains. The so-called *radiation boundary conditions at infinity* have been studied since 1912 by Sommerfeld, but its practical implementation in computational solutions of electromagnetic field propagation can be referred to Engquist and Majda [1977]. Its obvious that the occurrence of boundaries affects the evolution of a physical event that would otherwise propagate into open space. Many different strategies have been adopted to circumvent the problem. Among numerous approaches ,the method of *characteristics* is well exploited in the fixed grid numerical method framework (Poinsot and Lele [1992]). The *Perfectly Matched Layer* (PML) approach, that is, the use of an artificial absorbing layer, was devised by Berenger [1994] for simulation of electromagnetic waves and successively adopted in many wave field simulations: acoustics, seismic vibrations and fluid. The basic and simple idea behind the PML method is as follows: any wave pulse reaching a physical domain boundary, can be absorbed by adding a layer where sink per source terms are 'turned on'. These terms are usually multiplied by a coefficient varying from zero, within the domain, to a maximum value at the outer edge of the layer zone. The mathematical properties to be attributed to this zone can reach great accuracy and complexity, as shown in the work of Lin et al. [2011] on recent advancements for non-linear regime of the Euler equations to be adopted in the layer. Recently Modave et al. [2010] set up a simple but accurate PML method which has proved

useful for linear and non-linear shallow water simulations. Essentially, the same approach is taken here.
Summarizing, this chapter is focused on the study and implementation of a Smoothed Particle Hydrodynamics numerical code, in its weakly compressible variant (WCSPH), to deal with non-reflecting boundary conditions, starting from the Perfectly Matched Layer approach. Basically, the method exploits the concept of a physical damping which acts on a fictitious layer added to the edges of a computational domain. In this context is showed how using appropriate equations in the SPH algorithm, allows to carry out good results for both one and two dimensional cases of time dependent shallow waves propagating in a finite domain.

4.2 The absorbing layer method

In general, the model equations governing the fluid dynamics are rewritten adding a sink or source term to the original equations, as follows:

$$\frac{\partial A}{\partial t} = f\left(A, \frac{\partial A}{\partial x}, x\right) - \sigma(A - A_{out}), \tag{4.1}$$

where A is a generic fluid variable, $-\sigma(A - A_{out})$ is the corresponding sink or source term, A_{out} is the external boundary value and σ is the damping coefficient differing from zero in the damping region only. With an appropriate choice of the σ spatial function, this procedure yields extremely small reflection waves. All these techniques are used for fixed grids discretization of the equations. In the Lagrangian approach, the characteristic lines method has been suggested by Lastiwka et al. [2009] and Vacondio et al. [2012] which uses a simplifyed version of that procedure. Instead, the PML approach is by far simpler but, as far as the author of this thesis knows, it has not been studied in the context of a Lagrangian approach. Here, this strategy is adopted for the Lagrangian Smoothed Particle Hydrodynamics scheme and tested in the case of waves propagating in a finite tank, showing that the results are fairly good.

4.3 Lagrangian formulation of shallow water equations

The attention of this section is focused on the open boundaries problem for the shallow water waves in the SPH framework. The general gravity wave case has been accurately studied, but not in the context of boundary problems, by Antuono et al. [2011]. The governing equations, in conservative Eulerian form, for shallow

water waves, derived under the usual approximations of wave elevation much smaller than the full water depth and for constant bottom elevation, are well known:

$$\frac{\partial H}{\partial t} + div(H\vec{v}) = 0 \tag{4.2}$$

$$\frac{\partial H\vec{v}}{\partial t} + div(H\vec{v} \otimes \vec{v} + g\frac{H^2}{2}\mathbb{1}) = 0, \tag{4.3}$$

where H is the full water height, g and $\vec{v}$ are gravity and fluid velocity, respectively. De Leffe et al. [2010] also derived an SPH formulation slightly different from the one presented in this thesis.

4.3.1 The 1D shallow water case

To outline important physical elements in the absorbing zone, the simple 1D case was studied. When written in the Lagrangian form, we have for the wave height equation:

$$\frac{DH}{Dt} = -H(\frac{\partial v_x}{\partial x}), \tag{4.4}$$

where $\frac{D}{Dt}$ is, as usual, the comoving derivative. The equation of motion is:

$$\frac{Dv_x}{Dt} = -g\frac{\partial H}{\partial x} \tag{4.5}$$

This set of equations, can be formally satisfied by a *fictitious* fluid having a density $\rho \equiv H$ and a equation of state for the pressure defined by:

$$p = \frac{1}{2}g\rho^2, \tag{4.6}$$

so that the shallow water equations 4.4, 4.5 are fulfilled by this *special fluid* and therefore can be immediately approximated by the standard SPH formulae. Then, the SPH shallow wave equations are:

$$\frac{DH}{Dt} = -H\nabla\vec{v} \Rightarrow \frac{dH_i}{dt} = \sum_j m_k(\vec{v}_i - \vec{v}_j) \cdot \nabla_i W_{ij}, \tag{4.7}$$

where $m_k = H\triangle x_k$. For the momentum equation, a slightly different formulation was used. It produces more accurate results due to its higher sensitivity to the pressure gradient for this peculiar equation of state. This equation has been

used for instance by Riadh and Azzedine [2005] in their study of shallow water flows with SPH:

$$\frac{D\vec{v}}{Dt} = -\frac{1}{\rho}\nabla p \Rightarrow \frac{D\vec{v}_i}{Dt} = -\sum_j m_j \left(\frac{p_i + p_j}{\rho_i \rho_j} + \prod_{ij}\right) \nabla_i W(\vec{r}_i, \vec{r}_j) \tag{4.8}$$

Since $p = (1/2)g\rho^2$ and $\rho \equiv H$, then for shallow water momentum we have:

$$\frac{D\vec{v}_i}{Dt} = -\sum_j m_j \left(\frac{1}{2} g \frac{H_i^2 + H_j^2}{H_i H_j} + \prod_{ij}\right) \nabla_i W_{ij} \tag{4.9}$$

4.3.1.1 Test description and damping technique

In a domain of amplitude $X = 500$ it is produced a Gaussian pulse in the density profile and a corresponding fluid speed according to the following prescription:

$$H(x) = H_0 \left(1 + 0.01 exp\left(-\frac{(x - x_c)^2}{A^2}\right)\right), \tag{4.10}$$

$$v(x) = (H(x) - H_0)\sqrt{gH_0} \tag{4.11}$$

So we have a 1D *soliton* traveling towards the right side of the domain. The following pulse parameters have been chosen: $H_0 = 1$, $x_c = (3/4)X$ and $A = 9h$, where $h = 2$ is the interpolating particle size (the smoothing length). The particle spacing is $\triangle x = 1$. All quantities are in SI units.
A physical damping can, obviously, produce attenuation of outgoing signals; however, Fourier analysis shows that not all the harmonics belonging to a signal are attenuated and a special shape of the damping is needed, as developed for example by Modave et al. [2010] who made a good mathematical analysis that can be assumed to be valid also for the Lagrangian SPH method, at least in the shallow waves case, since the fluid speed is very small (essentially no mass transport) compared to the wave speed.
To damp appropriately the waves in proximity of the domain edge, an extra spatial layer is added to the domain and the equations in this damping layer are:

$$\frac{DH}{Dt} = -H\frac{\partial v_x}{\partial x} - \sigma(x)\left(H(x) - H_0\right), \tag{4.12}$$

for the water level, and

$$\frac{Dv_x}{Dt} = -g\frac{\partial H}{\partial x} - \sigma(x)\left(v_x - v_0\right) \tag{4.13}$$

for the fluid speed, where $\sigma(x)$ is the damping coefficient, which is a function of the position in the layer, having an appropriate spatial dependence (discussed below), and v_0 is the outflow speed along x-direction. In this study it is imposed to be $v_0 = 0$, since this is a case in which waves are propagating in a closed water tank. To produce a suitable damping layer, the following terms are added:

1. $S = -\sigma(x)(H - H_0)$ to the density equation.
2. $Q = -\sigma(x)(v_x - v_0)$ to the momentum equation.

For the coefficient σ, the following functions, suggested by Modave et al. [2010], were tested:

$$\sigma = \sigma_0 \left[\frac{(x - x_0)}{L}\right]^m, \tag{4.14}$$

and

$$\sigma = \sigma_0 \left[\frac{(x - x_0)}{(x_0 + L) - x}\right], \tag{4.15}$$

where the exponent m is a positive integer to be tuned. The parameters x_0 and L are respectively the starting point and the thickness of the absorbing zone. Furthermore, also two *ad hoc* treatments were tested. They were called *switches*, based on the following physical intuition:

a. decrease the horizontal pressure force, only in the damping layer, with particular functions f_1 or f_2, that is

$$f_1(x) = \begin{cases} -\frac{x-(x_0+L)}{L} & x_0 < x < x_0 + L \\ 1 & x \leq x_0 \end{cases} \tag{4.16}$$

and

$$f_2(x) = \begin{cases} \frac{L^2-(x-x_0)^2}{L^2} & x_0 < x < x_0 + L \\ 1 & x \leq x_0 \end{cases} \tag{4.17}$$

f_1 is a linear function, while f_2 is parabolic one with its maximum at x_0. The forces on the particles are simply multiplied by these functions called, in this work, *cutting functions*, since they reduce to zero the horizontal force acting on particles close to the end of the damping layer. In this case, the basic physical idea is to decrease smoothly the pressure gradient near the boundary.

b. Use the damping friction $\sigma > o$ only if $v_x < 0$. In this case, the idea is to damp preferentially fluid motion directed into the domain.

Fig.4.1 shows the initial analytical configuration for the fluid speed v_x (not the wave speed) and both the totally reflected Gaussian and the undisturbed propagated wave (here called *infinity case*), both computed at the time $t = ((X/2)/\sqrt{gH_0}))$. The term "*infinity*" means the remnant of the numerical solution in the computational domain when the corresponding peak of the analytical solution went out of the integration domain. It is obtained by simply integrating the numerical solution into a much larger numerical domain, obviously consuming more CPU time. The total reflection profile is due to the wave coming back at the perfectly rigid boundary located at the right domain edge $X = 500m$.

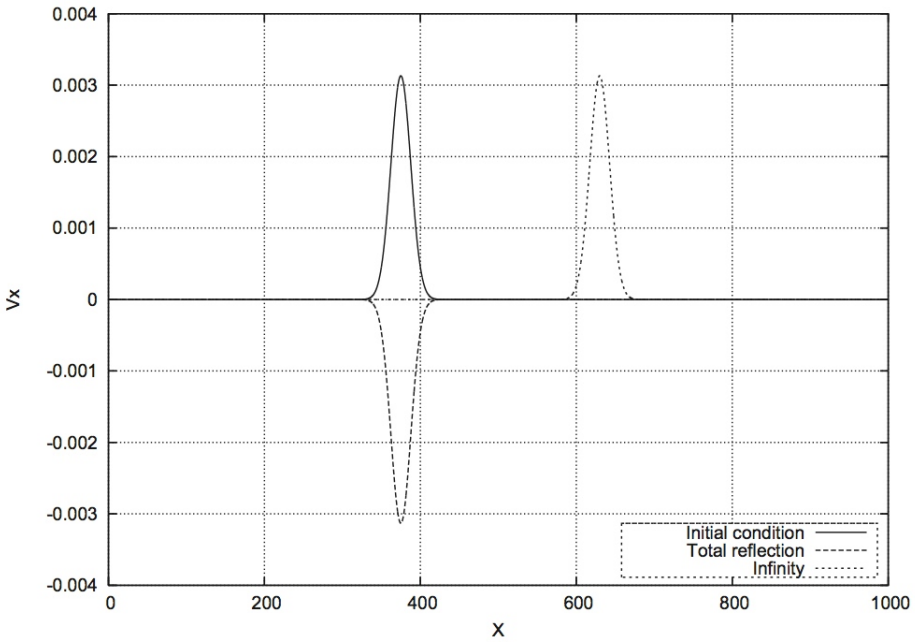


Figure 4.1: Initial pulse profile with chosen parameters $H_0 = 1$, $x_c = (3/4)X$ and $A = 9h$: solid line; the perfectly reflected pulse: dashed line and the infinity case: dotted line.

The reflection ratio R for this set of $1D$ simulations is computed following the Modave et al. [2010] formula, i.e. the ratio of the errors

$$R = \sqrt{\frac{E_{lay,\infty}}{E_{refl,\infty}}}, \tag{4.18}$$

where $E_{refl,\infty}$ is given by:

$$E_{refl,\infty}(t) = \frac{1}{2}g \int (H_{refl} - H_\infty)^2 \, dx + \frac{1}{2}H \int (v_{refl} - v_\infty)^2 \, dx \tag{4.19}$$

and the label ∞ identifies the values obtained with an extremely far right edge, i.e. no boundary condition (BC), the label lay refers to the quantities evaluated with a specific absorbing layer thickness and the label $refl$ refers to the quantities evaluated with a totally reflecting BC. Obviously the integrals have been replaced by a sum over the particles. Essentially, it is measured the differences of the flow variables and then the relative energy is computed, that is, it is not computed the differences of the energies contained in the integration domain, that would be:

$$\frac{1}{2}g \int \left(H_{lay}^2 - H_\infty^2\right) dx + \frac{1}{2}H \int \left(v_{lay}^2 - v_\infty^2\right) dx$$

The time evolution of the wave height, for a $100m$ amplitude layer with a hyperbolic damping function (Section 5.1), typically appears as depicted in Fig.4.2, where each profile is taken at time intervals of $\triangle t = 7.4s$.

4.3.2 Waves in a tank: 2D case

This case is focused on the wavy motion produced by a wave maker palette in a water tank. The dynamics is truly two dimensional. To simulate incompressible water waves, the weakly compressible approximation of SPH (WCSPH) was exploited. As mentioned before, it consists essentially in the use of a sound speed an order of magnitude larger than the maximum typical speed of the water. The choice of the parameters was $c_{sound} = 20 v_{typical}$, where $v_{typical} = \sqrt{gH}$. It is noteworthy that, as suggested by Madsen and Shaffer [2006], here is used as typical speed the wave speed and not the very small fluid speed. The governing equations are the ones seen in eqs. 4.4, 4.5 (extended to the two-dimensionsl case), but with the equation of state given by the Tait equation 2.45.
The wave maker is placed at the left side of the rectangular tank. A damping layer is added in the right side. Check points of the water level are defined at regular space intervals. The setup is shown in Fig.4.3.

The aim is to produce, in the finite tank of length X, a motion unaffected by the right boundary, i.e. equal to the one obtained in the same zone, but in an infinite tank. It was added the same source terms used for the $1D$ case to the equations of motion and continuity, taking into account the dimensionality of the problem. The term $Q_\alpha = -\sigma_\alpha \left(v_\alpha - v_{0_\alpha}\right)$ to each component of the momentum equation ($\alpha = x, y$), where v_{0_α}, as the $1D$ case, is set to zero. Also in this case,

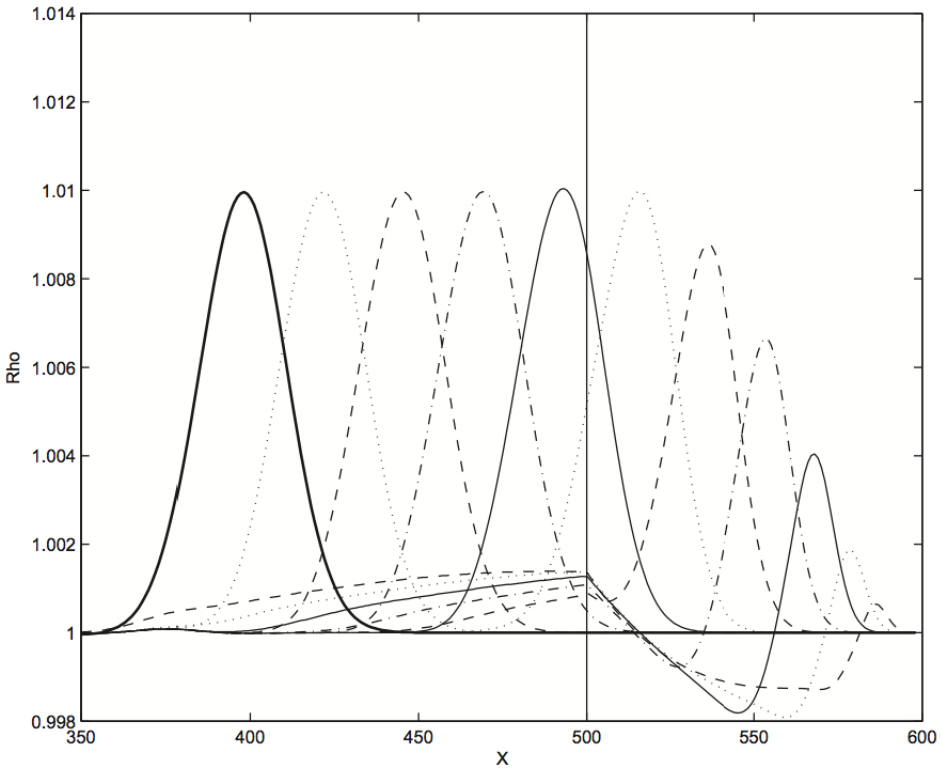


Figure 4.2: Time evolution of the wave height at intervals of $7.4s$; the bold solid line represents the initial configuration.

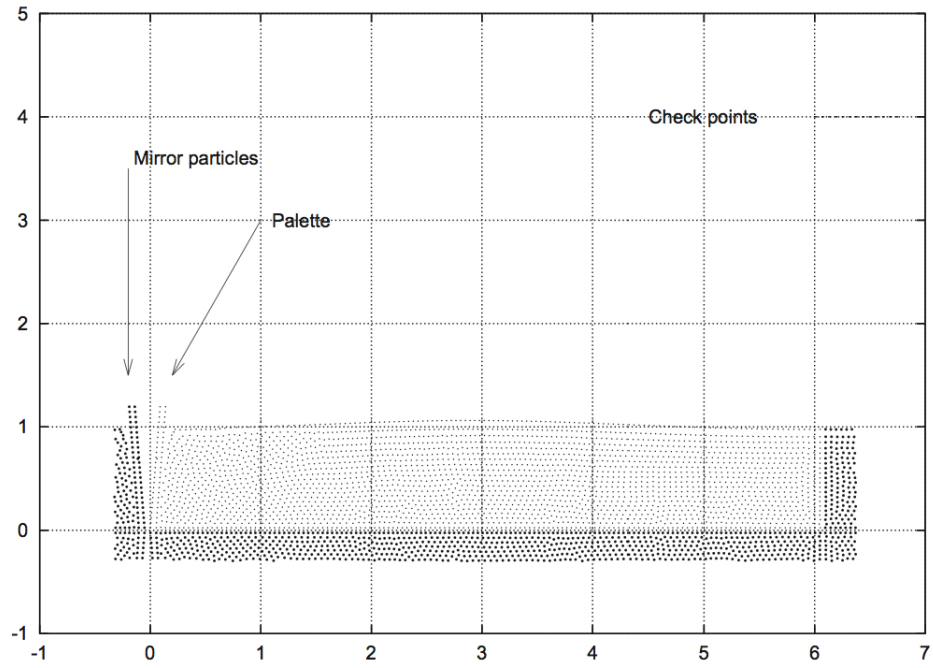


Figure 4.3: Tank with wavemaker and mirror particles, with no layer added.

some "*ad hoc*" terms, guided by the $1D$ experience and physical intuition were explored:

a. a switch on the damping triggered by the speeds v_x and/or v_y,

b. the use of cutting functions to reduce the horizontal component of the force due to the pressure.

The cases under consideration in this thesis were those of a continuos periodic wave and a wave generated by a single sinusoidal oscillation.
The next chapter will be devoted to the presentation of the simulation results of the above mentioned cases.

Chapter 5

1D/2D Shallow water SPH simulation results

5.1 1D simulation results: Gaussian pulse

Fig.5.1 shows the reflection ratio for the various damping functions obtained with different values of the exponent m (see Section 4.3.1.1) and the hyperbolic function 4.15.

For small layers, the best performances were obtained for the linear $m = 1$ and the hyperbolic functions. However the study was focused on the hyperbolic function only, since it shows better results when coupled with the *ad hoc* physical switches. Fig.5.2 shows the reflection ratio as a function of the thickness of the absorbing layer for two different functions of the absorption (left panel: $m = 1$, right panel: hyperbolic), i.e. it is a comparation between functions which are more efficient when it is the small absorbing layer case.

The solid line identifies the case with the use of the function without any switch, called here *clean* function; the dashed line identifies the results obtained with the same clean function with the addition of a further tool: attenuation of the pressure force with a linear function f_1. The dotted line identifies the case of attenuation of the pressure force with a quadratic (parabolic) function f_2. The best results are obtained with the dashed line, that is, with the linear cutting function. Fig.5.3 shows the reflection ratio results obtained by adding the velocity switch. The best results are displayed with a dotted line, corresponding to a damping with the cutting function f_1 and with simultaneous use of unidirectional friction, i.e. use of the damping friction σ only if $v_x < 0$, so that the damping acts only if the speed of the particles (not of the wave) is negative.

Fig.5.4 compares the best results obtained with the exponent $m = 1$ and the hyperbolic damping function, using both switches f_1 and $\sigma \neq 0$ if $v_x < 0$.

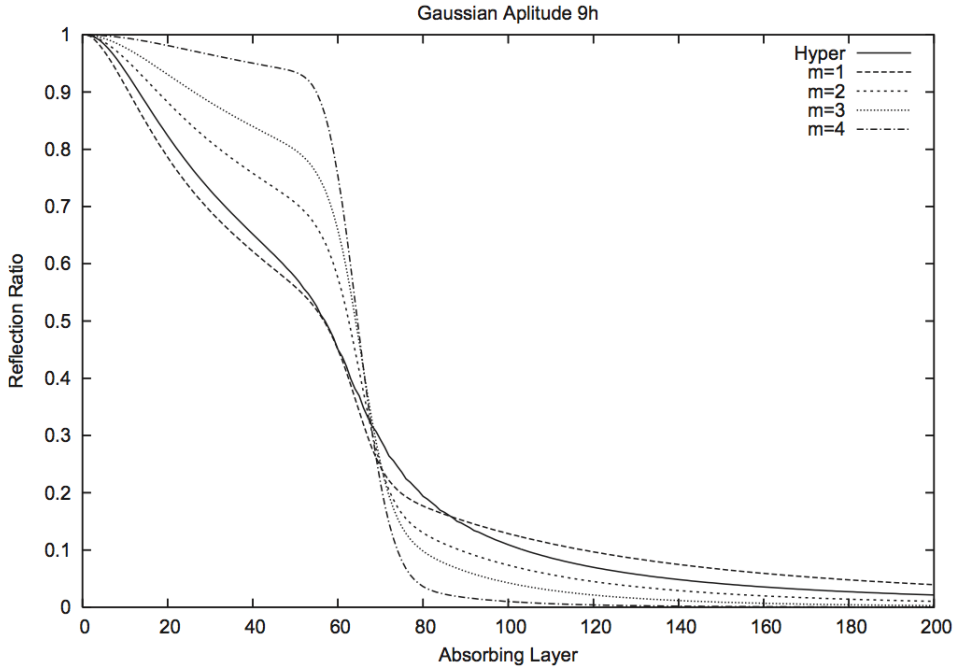


Figure 5.1: Reflection ratio versus the thickness of the absorbing layer for σ function, hyperbolic and power law function with different values of the exponent m. The parameters are identical to those used in Fig. 4.1.

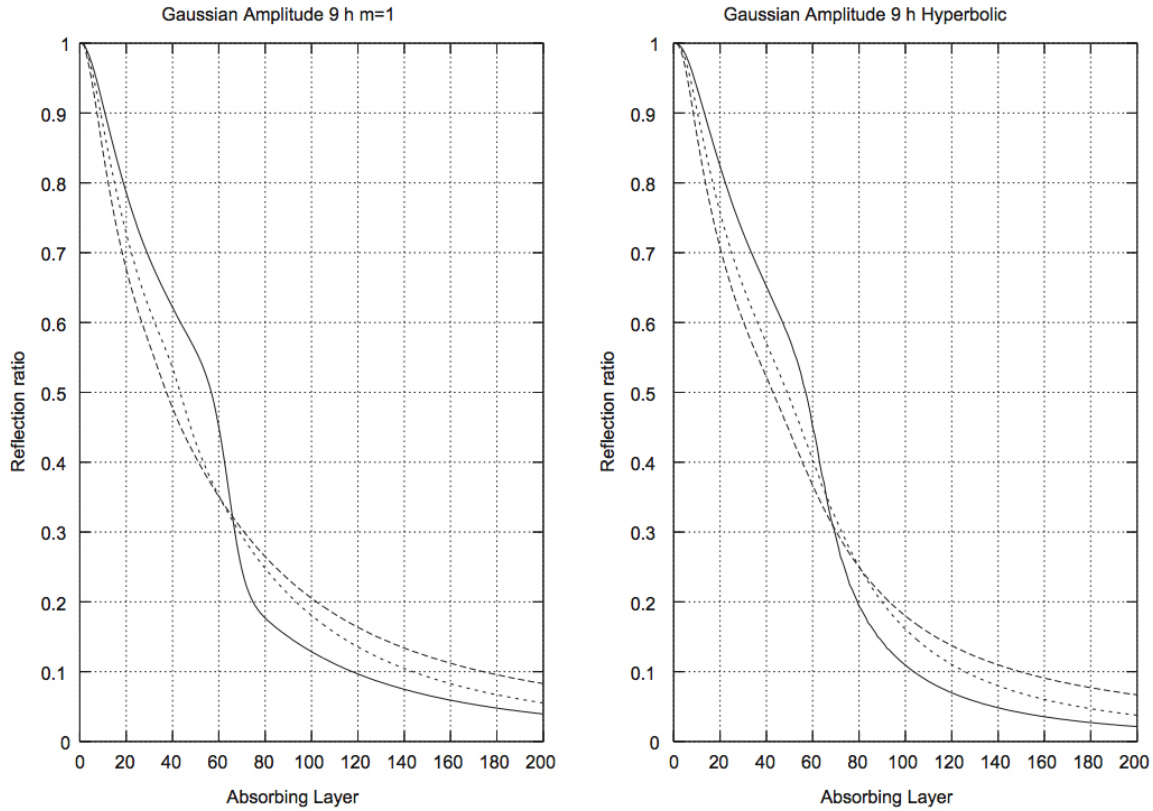


Figure 5.2: Reflection versus the thickness of the absorbing layer for two different absorbing coefficient functions (left panel: $m = 1$, right panel: hyperbolic). The solid line identifies the case with the use of the *clean* function only and the dashed line identifies clean function multiplied by f_1. The dotted line identifies the clean function multiplied by f_2. The parameters are identical to those used in Fig.4.1.

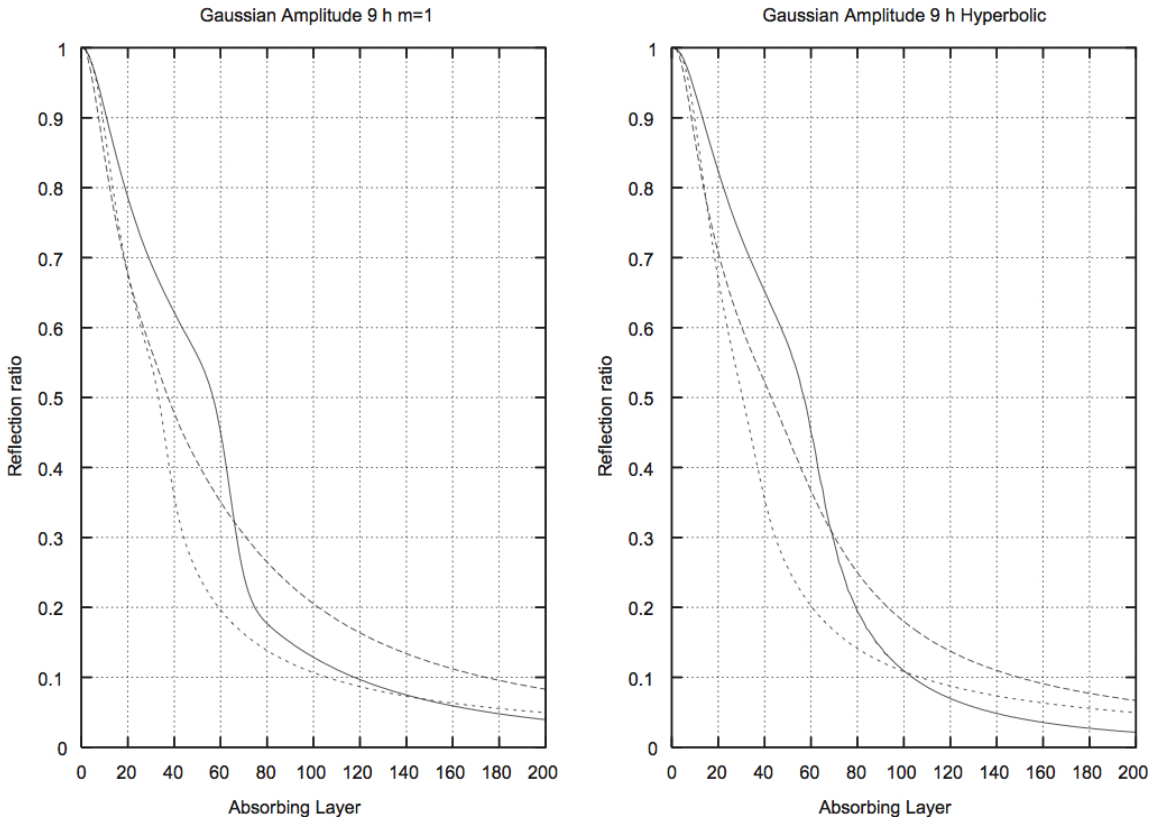


Figure 5.3: Reflection ratio versus the thickness of the absorbing layer for two different distributions of the absorption coefficient (left panel: $m1 = 1$, right panel: hyperbolic). The solid line identifies the case with the use of the *clean* function, and the dashed line identifies the same case with the attenuation of the force added with a linear function f_1 within the layer zone. The dotted line identifies the previous case with a further switch on the speed. The parameters are identical to those used in Fig.4.1.

The hyperbolic function works only moderately better, but in the $2D$ case it was observed a much better performance and therefore the attention was focused on that function.

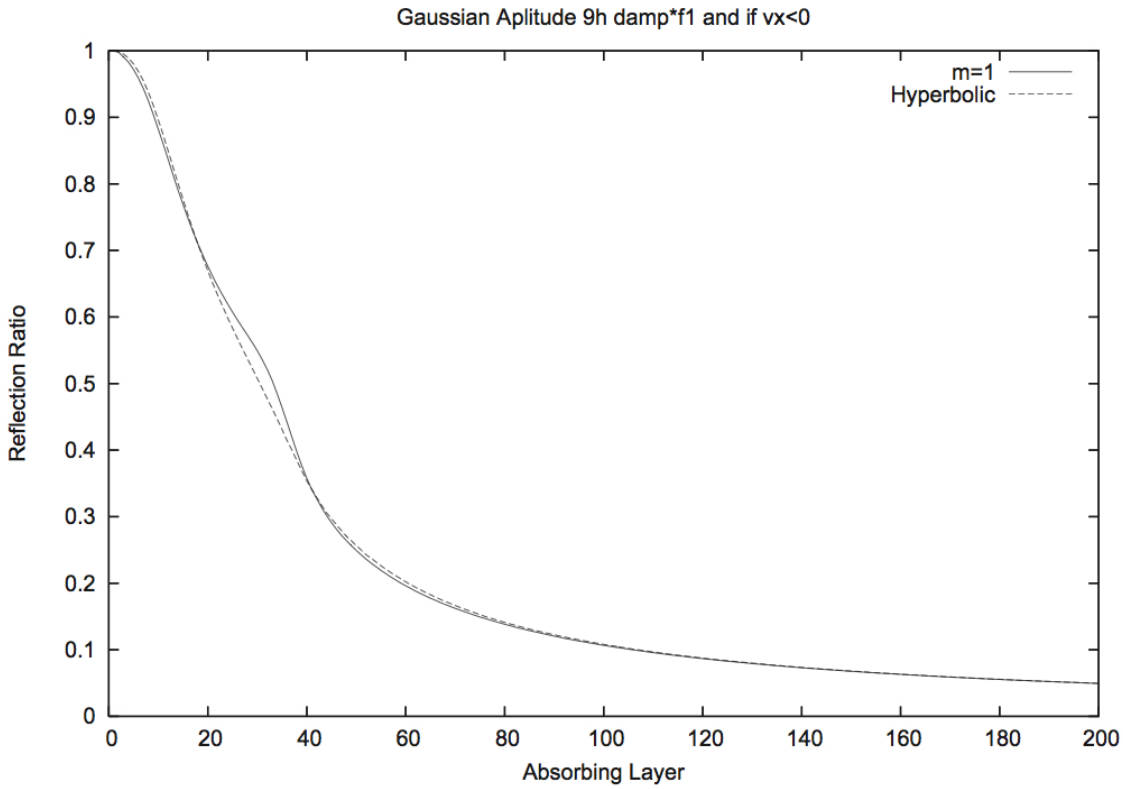


Figure 5.4: Reflection ratio versus the thickness of the absorbing layer for two different distributions of the absorption coefficient (solid line: $m = 1$ and dashed line: hyperbolic). With attenuation of the horizontal pressure force with a linear function f_1 within the damping zone and use of damping only if the speed of the particles (not of the wave) is negative $v_x < 0$ in the absorbing layer. The parameters are identical to those used in Fig.4.1.

5.2 2D simulation results: waves in rectangular tank

5.2.1 Continuos periodic wave

A rectangular tank of length $X = 6.061$ meters and height $1m$, was generated. The particles had an intrinsic width $h = 0.1$ and were placed with a regular spacing $\triangle l = 0.05$ along X and Z. The number of particles in the tank was $N = 2570$ (wavemaker included). The boundaries are made with mirror particles procedure. The wave maker oscillates with a period $T = 2.23s$ and with an angle amplitude of 5°. With these values, the water in the tank enters in a resonant state. Fig.5.5 reports the levels of the water column, measured at five different positions (at $x = 1, 2, 3, 4, 5m$) along the tank, versus elapsed time, in the case of an "infinite" or very long tank. The levels are vertically shifted for clarity. It is clear that the waves propagate without disturbances.

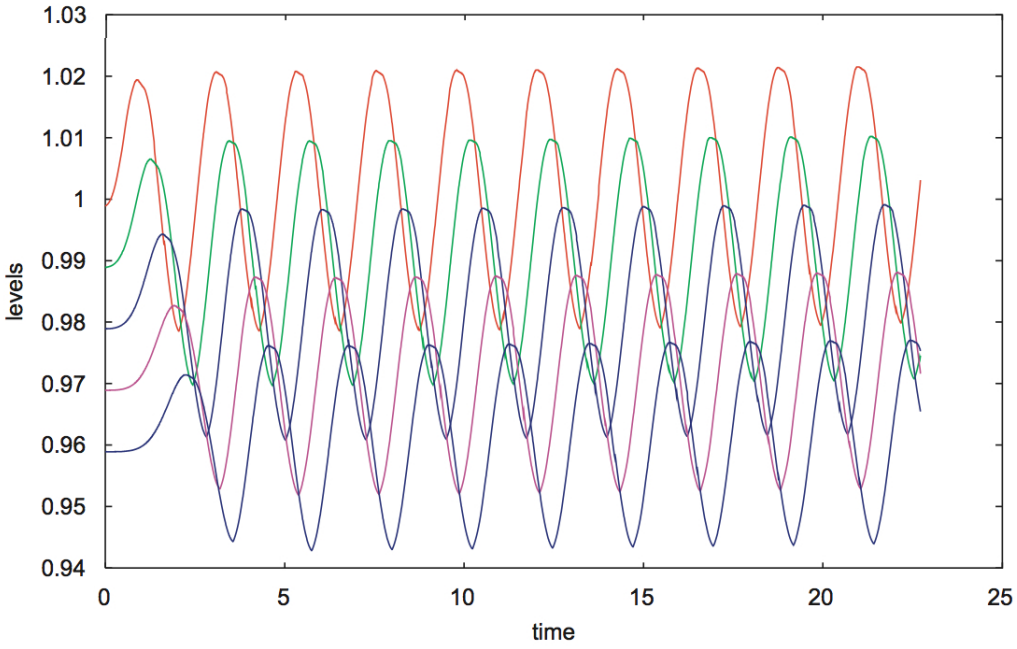


Figure 5.5: Levels at $x = 1, 2, 3, 4, 5m$ in an 'infinite tank'.

Fig.5.6 shows the velocity field of fluid in the tank in resonant condition, with the wavemaker an mirror points.

Fig.5.7 shows the levels for the resonant tank. The levels are shifted in the Z coordinate by a small amount $dz = 0.01$ for clarity. It is clear that the oscillations

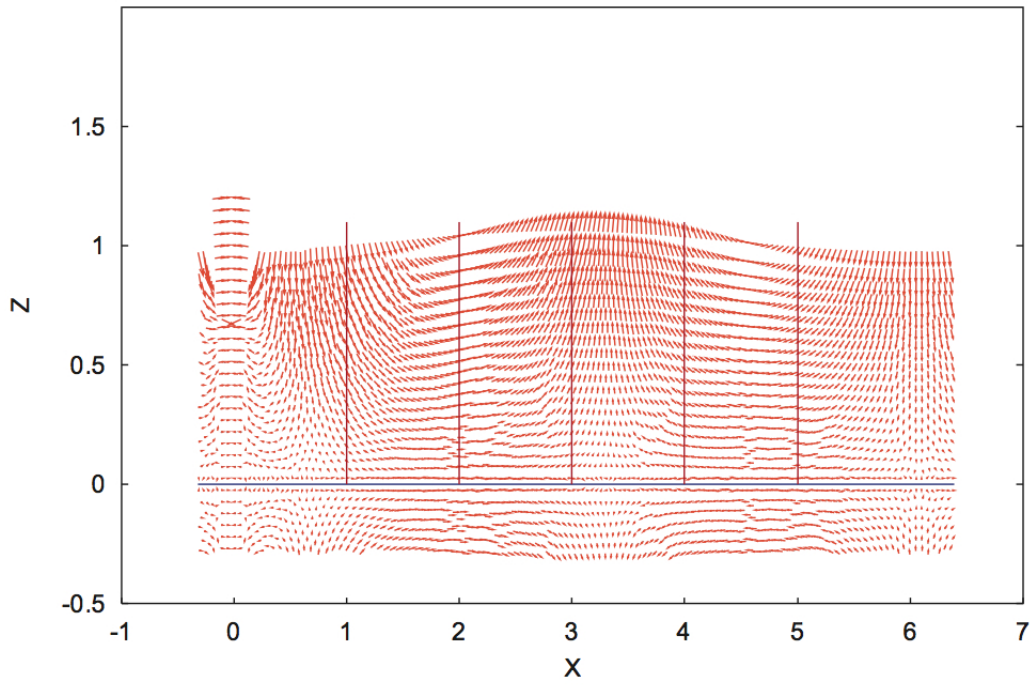


Figure 5.6: Velocity field of the water in the resonant case, with no damping layer.

are larger and increasing with time.

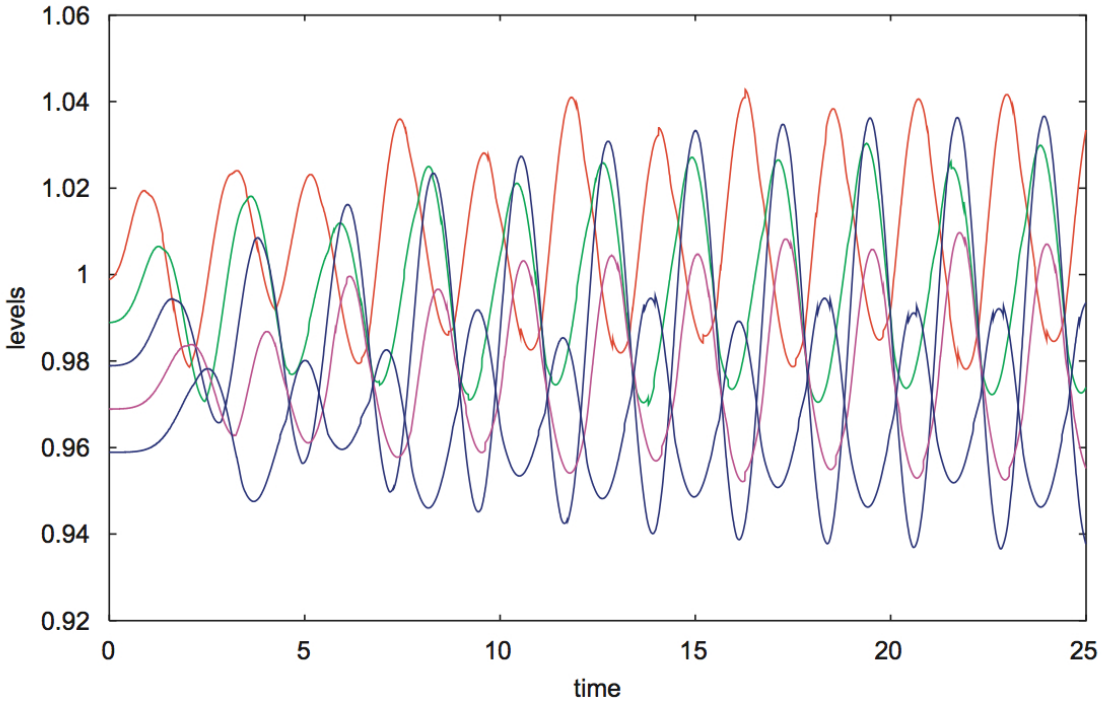


Figure 5.7: Water levels in the resonant case.

In Fig.5.8 are showed the resulting levels when a damping layer of extension $L = 6.061m$ is added. That is, it was used a full simulation domain $X = 12.122m$. In all these tests the hyperbolic function $\sigma = \sigma_0 \left[(x - x_0) / ((x_0 + L) - x + 0.5h)\right]$ is adopted. The coefficient σ_0 has the dimension of 1/time and its value is chosen such as $\sigma_0 = v_{ref}/L$. For this study the reference speed has been chosen equal to the sound speed: $v_{ref} = c_{sound}$ (case with the label M_0).

It is clear that the wave profiles are very similar to the ones of the *infinity* case.The kinetic energy content can be used as an indicator of the similarity of the flows. Fig.5.9 shows the kinetic energy of the water (computed excluding the damping layer contribution) versus time, for various cases. The increase of the energy in the resonant condition, the steady oscillating energy for the *infinity* case, together with the very close values obtained with different damping layer cases are clearly shown.

If Fig.5.9 is enlarged, zooming the values of the kinetic energy, Fig.5.10 is obtained. It allows to find some discrepancies between the *infinite* case and the M_0 one. It was made the same simulation, to test the following set of switches:

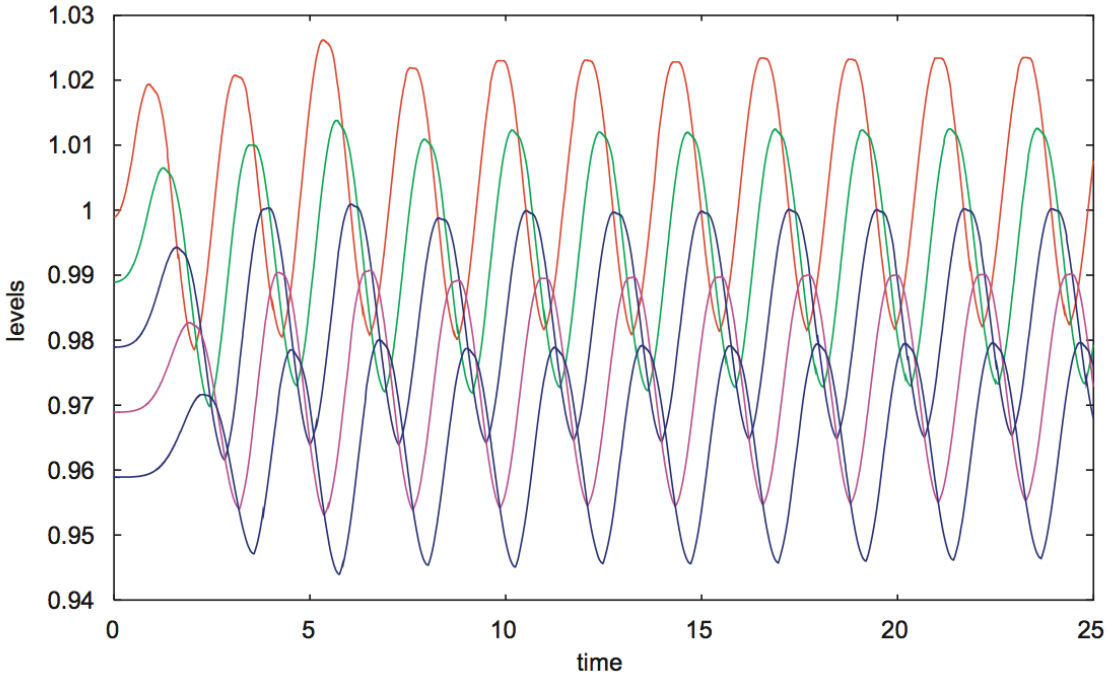


Figure 5.8: Water levels for the case M_0: plain hyperbolic absorption function.

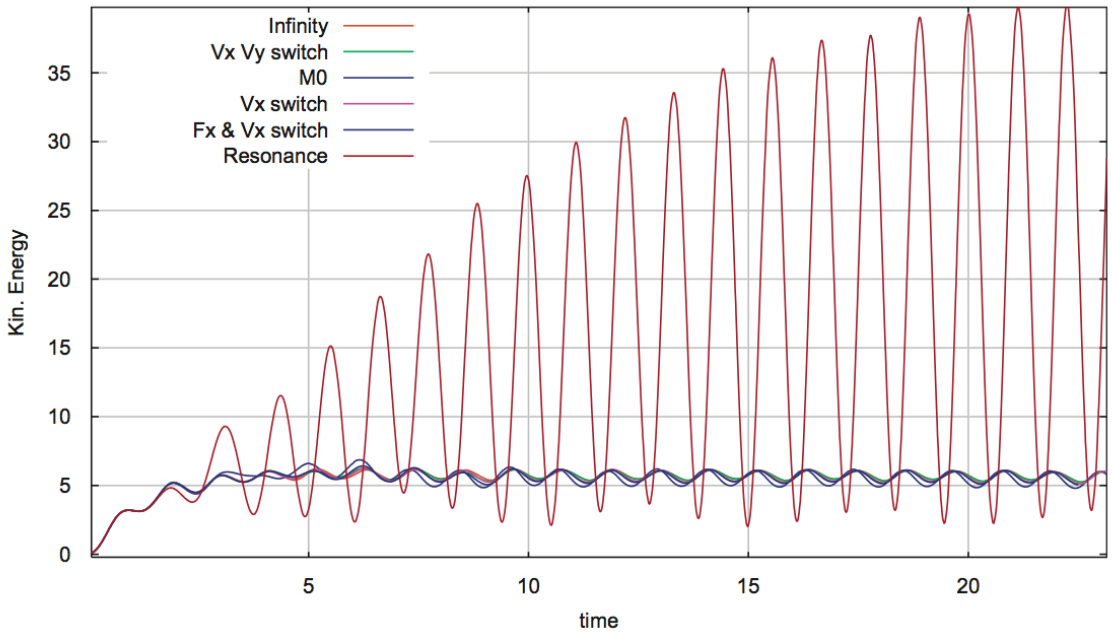


Figure 5.9: Kinetic energy versus time for resonance and for boundary layer with switches.

V_x, V_y : the damping functions works only if $v_x < 0$ and $v_y < 0$,

V_x : the damping functions works only if $v_x < 0$,

F_x, V_x : the damping functions works only if $f_x < 0$ and $v_x < 0$.

Taking a look look at the kinetic energy, it can be observed that performance better than M_0 are obtained with each of the switch options mentioned, since for M_0 the kinetic energy has oscillations larger than the ones of the *infinity* case. Qualitatively, the best results seems to be obtained with the simple 'V_x' switch. However further investigations should be carried out to produce a numerical estimate.

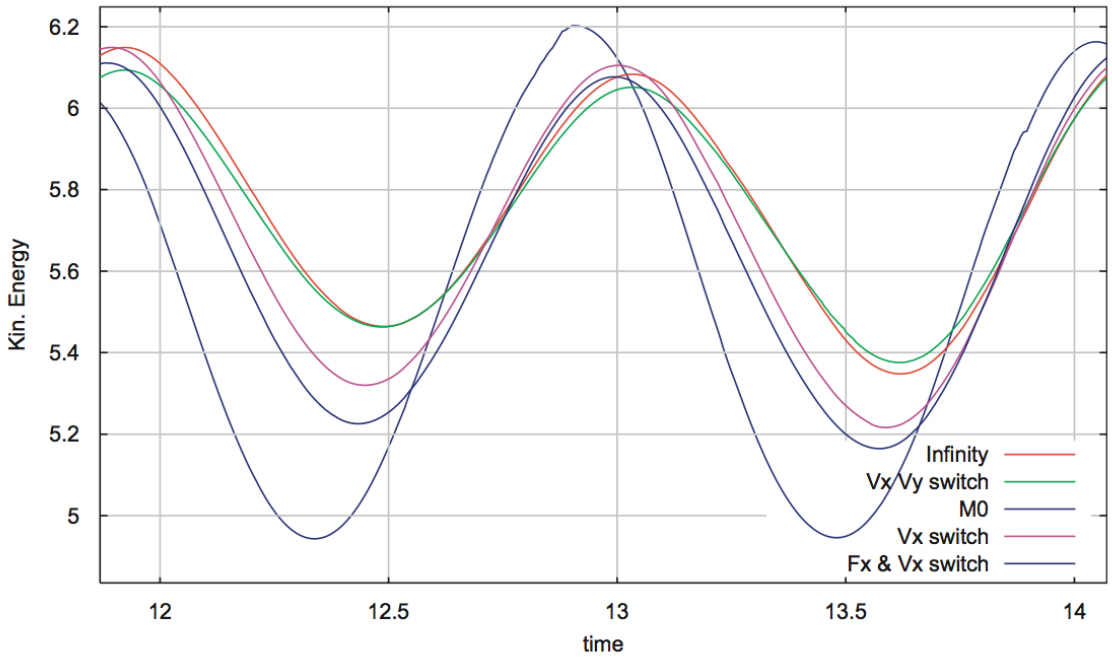


Figure 5.10: Zoom on the kinetic energy versus time.

5.2.2 Sinusoidal single impulse wave

The simulation of shallow water waves was made also for a single impulsive sinusoidal wave. In the same tank, the wavemaker makes a single oscillation with the same angular amplitude and period of the previous simulation. In this case it is easy to see the effects of the reflected wave. To have a detailed information,

ten elevation level points were created and located at intervals of $0.6m$, starting from $0.3m$ from the left side. Fig.5.11 shows the particles and their velocity field for the water in the tank at *time* $t = 25s$; this is the configuration without any damping layer, i.e. clean reflection conditions at the right side. The vertical lines identify the points of level measurement. The particles distribution and the speed arrows show that the water level is still oscillating.

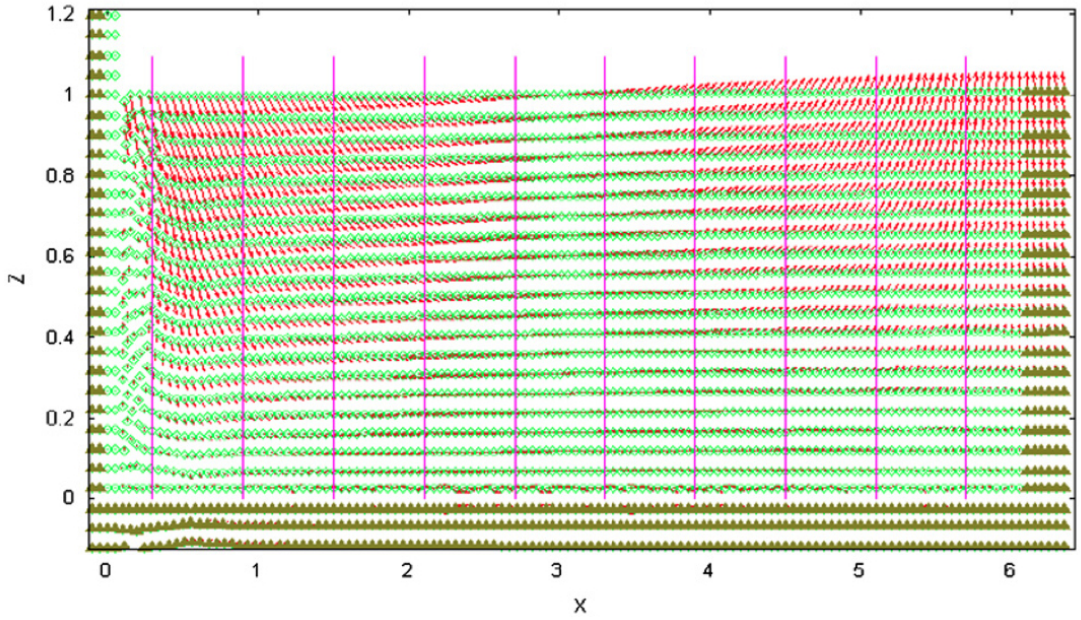


Figure 5.11: The particles configuration and their speed (enlarged by a factor 2.5) for the water in a reflecting tank.

Fig.5.12 shows in the same panel the levels when the tank has a simple reflecting boundary on the right side; the levels obtained for the *infinite* tank (the thicker and straighter horizontal lines), are also plotted. It is obviously clear that oscillations are present even after the long time the pulse had to be outside the tank.

In Fig.5.13 are shown the levels obtained with the M_0 prescription compared with the *infinity* case. In Fig.5.14 the M_0 levels are plotted together with the ones obtained with the 'V_x, V_y' switch.

It was also studied the same two problems using smaller damping layers. The results are similar to the one presented here with the obvious difference that the damping effects diminish as the layer thickness decreases. For what concerns the results of the single sinusoidal impulse, Fig.5.15 shows the kinetic energy of the

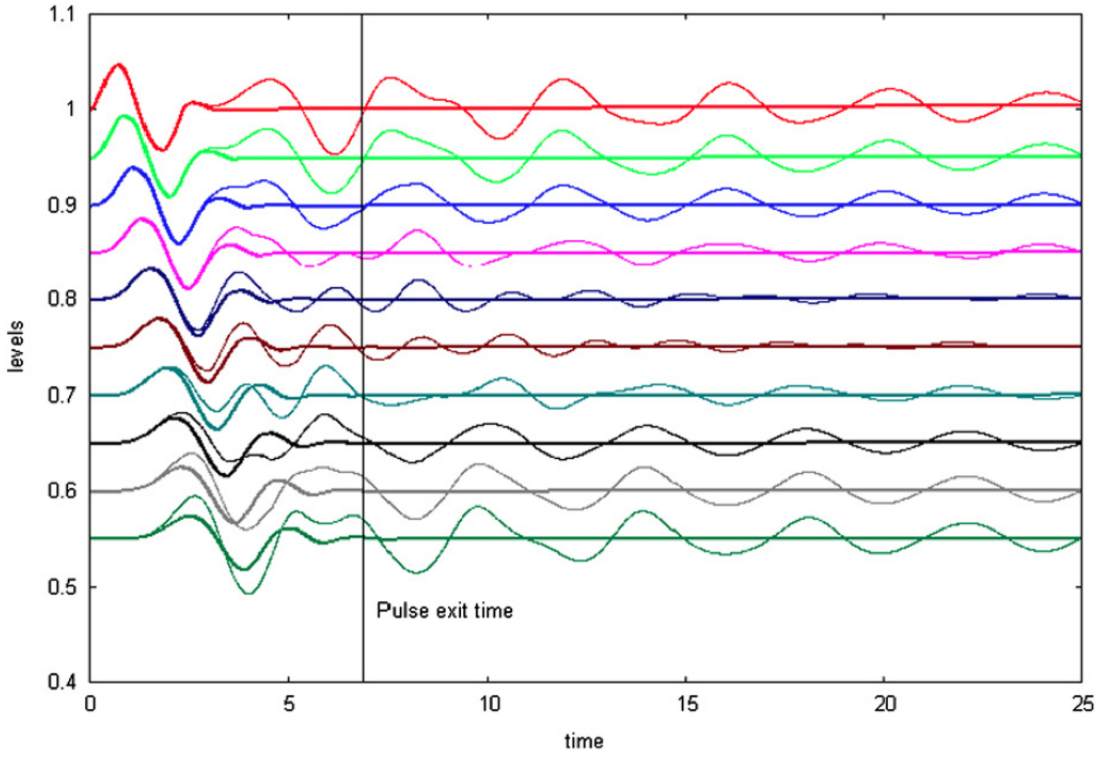


Figure 5.12: Ten levels for reflecting BC tank compared to the levels of the infinity case.

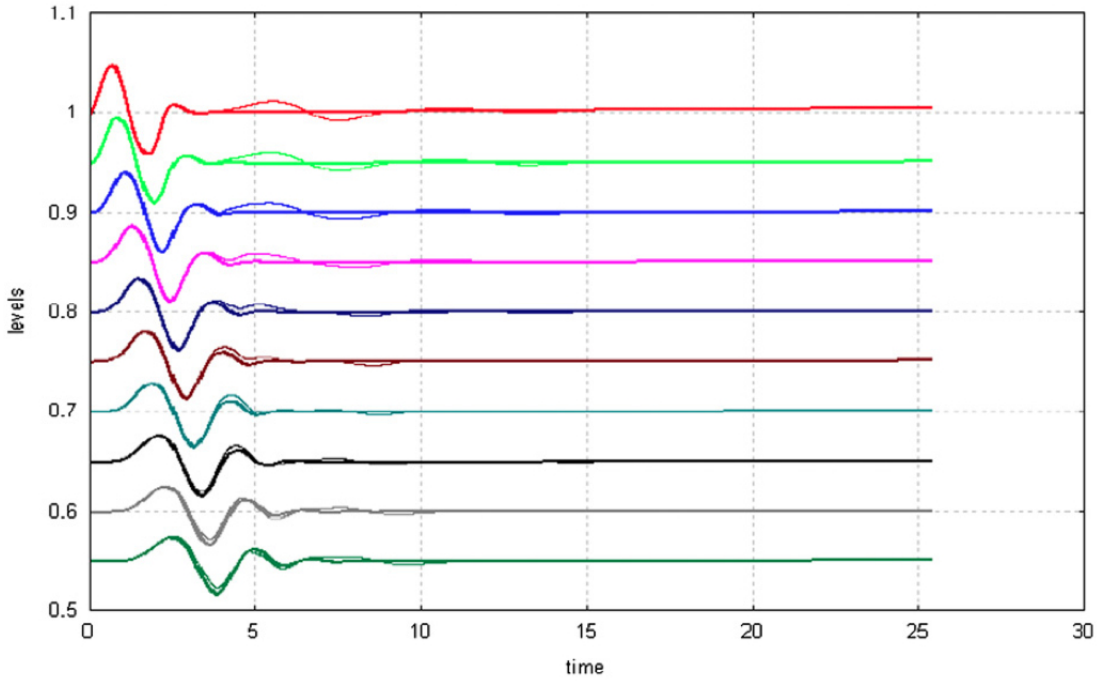


Figure 5.13: Comparison of levels obtained with the M_0 prescription and the infinity case.

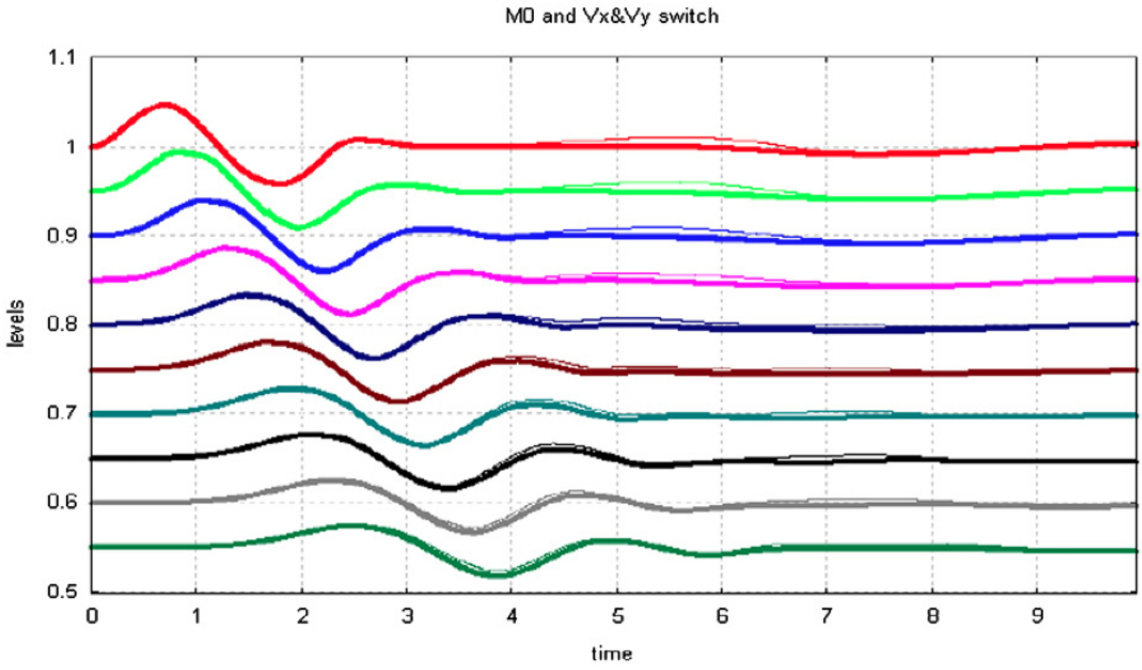


Figure 5.14: Comparison of levels obtained with M_0 and with the switches V_x and V_y (thick lines).

tank versus time when the length of the damping zone is $L = 6.06m$.

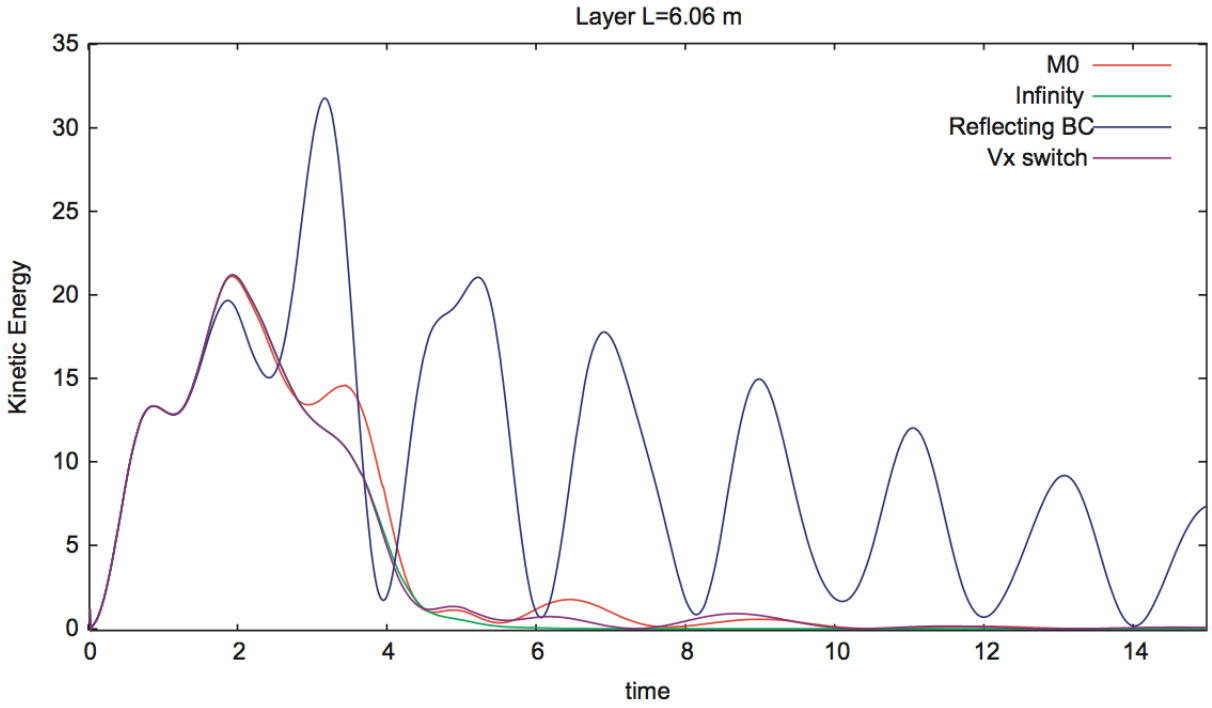


Figure 5.15: Kinetic energy for different damping actions in the case of a $6.06m$ large damping zone.

Fig.5.16 shows the kinetic energy of the water in the tank for a shorter damping layer $L = 3.03m$. It was also tested the use of cutting functions, but the improvements are very small to be appreciated in the figure.

From Fig.5.16 it is clear that the reducing action of the residual oscillations is due to the velocity switch when added to the plain damping function. Fig.5.17 shows a zoom of Fig.5.16, to show clearly the different effects of the damping criteria. It shows that, for the $2D$ problem, the cutting function improves the results over the plain damping, but it is not better than the simple velocity switch. the joint action of the velocity switch and of the cutting function does not improve the results.

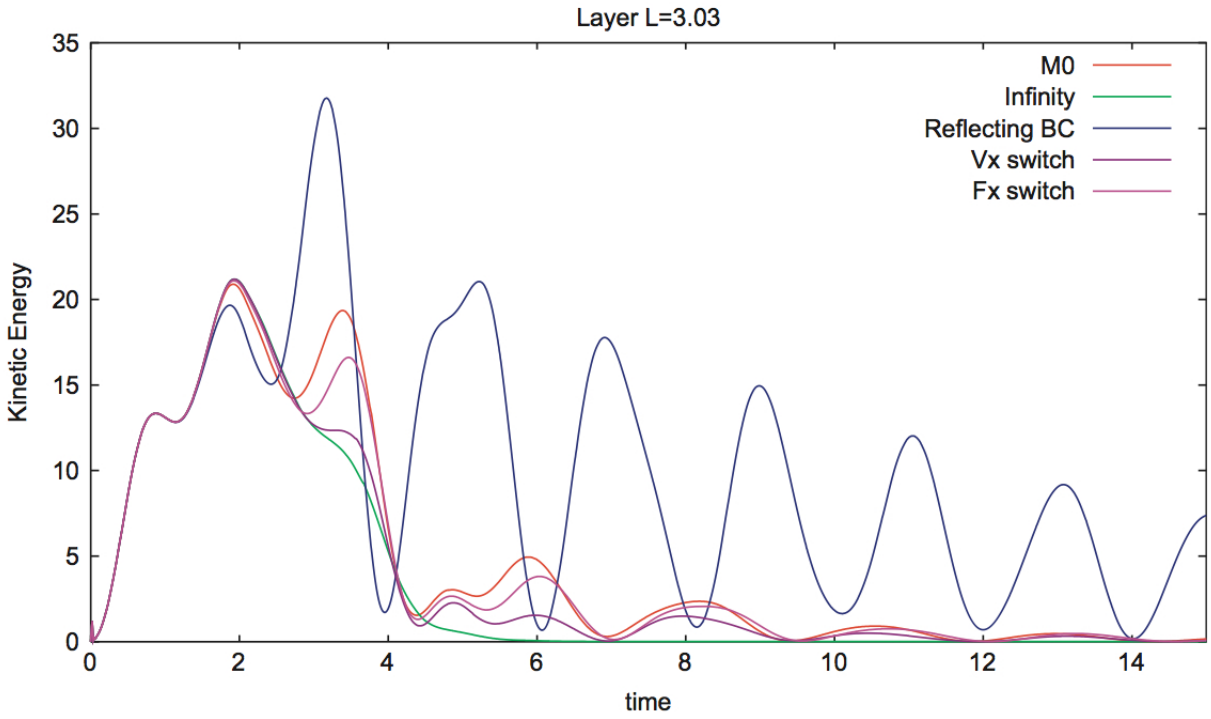


Figure 5.16: Kinetic energy of the water in the case of damping layer $L = 3.03m$.

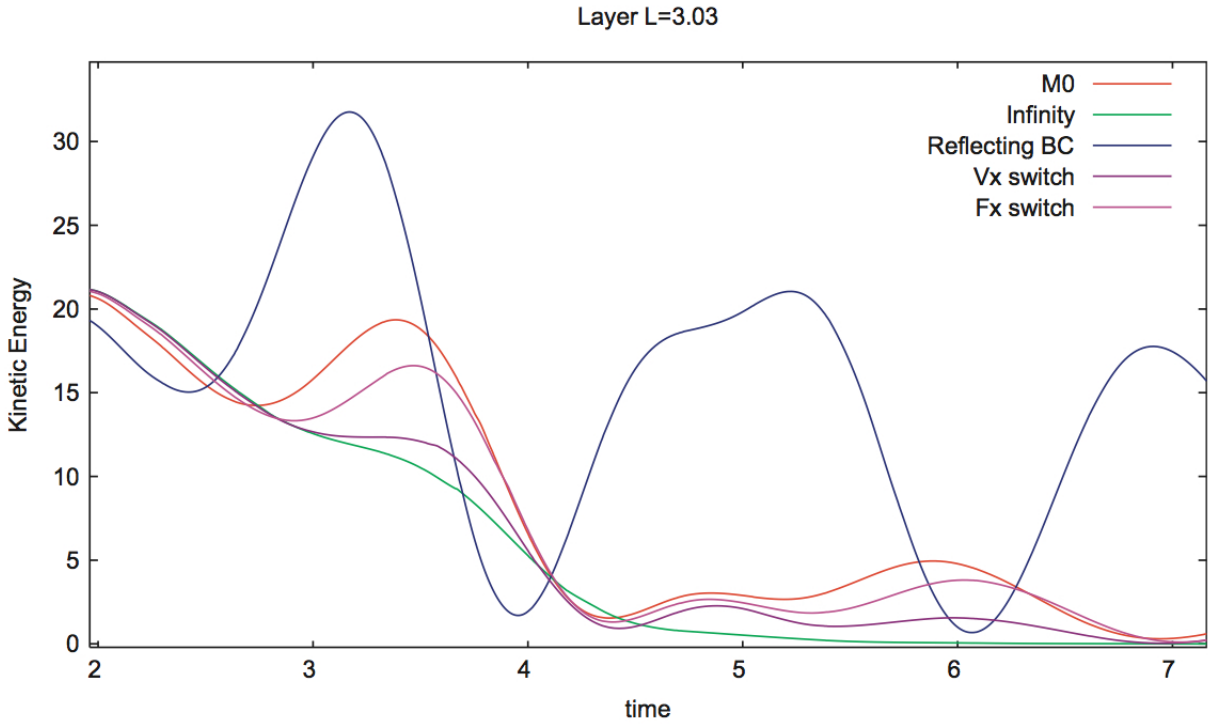


Figure 5.17: Zoom of Fig.5.16 to show better results of the different damping algorithms.

Chapter 6

Conclusions

In the context of this work , a numerical code has been implemented with the meshless technique named SPH (Smoothed Particle Hydrodynamics). The code was developed in its two variants : weakly compressible (WCSPH) and incompressible (ISPH). Both versions have been optimized and tested in standard cases and subsequently validated by comparison with experimental data, when available. In all cases addressed, the developed code has provided good results, particularly in the dam break and shear cavity problems, both in the pure two-dimensional, or three-dimensional case. In the dam-break simulation, the comparison with experimental data retrieved from the work of Martin and Moyce [1952], the results obtained seem to be even better than those obtained in several recent papers on this topic. In some cases, the results produced by the Lagrangian SPH technique were compared with those obtained by the Eulerian *grid-based* consolidated techniques and scarcely appreciable discrepancies have been found. A fundamental purpose of this part of the work is to lay the groundwork for a future development of a *hybrid* numerical code for the solution of fluid dynamics, where the fixed Eulerian grids and free Lagrangian nodes (particles) can be used simultaneously within a computational domain. The basic idea is to develop a method that can take advantage of either technique, where they are more suitable as, for example, in the determination of the free surface of a fluid, which is naturally obtained with the Lagrangian approach or in the analysis of extended domains, in which the Eulerian approach requires lower computational costs.
As regards the part of the work on the shallow water waves simulations, resuming the $1D$ case, it can be said that the best results have been obtained with the hyperbolic damping function $\sigma_0 \left[(x - x_0) / ((x_0 + L) - x) \right]$, plus two further treatments: the decrease of the horizontal pressure force with a linear function f_1 only in the damping layer and the use of damping only if the speed of the particles in the absorbing layer is negative, that is $v_x < 0$.

It must be emphasized that, since a Lagrangian approach was used (the particles are free to move), it was added in the denominator an extra softening term $0.5h$ to avoid division by zero if a particle reaches the left edge:
$\sigma_0 \left[(x - x_0) \,/\, ((x_0 + L) - x + 0.5h) \right]$. So hereafter it is reported the results obtained with hyperbolic damping function only. Finally, Fig.6.1 shows the values of the reflection coefficient versus the amplitude of the damping layer for different widths of the Gaussian pulse. It shows the predictable result that the increase of the amplitude requires a larger damping layer to obtain the same reflection ratio. Regarding to the $2D$ shallow waves in a tank, the use of a damping layer is successful in avoiding boundary reflections into the computational domain. An obvious requirement is that the absorbing layer must be greater than or equal to the maximum significant wavelength produced by the physical simulation.
Both in the $1D$ and $2D$ cases, the basic procedure can be improved by the use of appropriate switches. A simple and efficient switch is the one that makes the damping operate for negative speeds only, i.e. $v_{x<0}$. The switch that reduces the horizontal component of the force, F_x, is efficient in $1D$, but not so much for the $2D$ problem examined. Further work is in progress to make a quantitative evaluation of the $2D$ simulations and verify the affordability of the method in the case of highly compressible fluid dynamics.

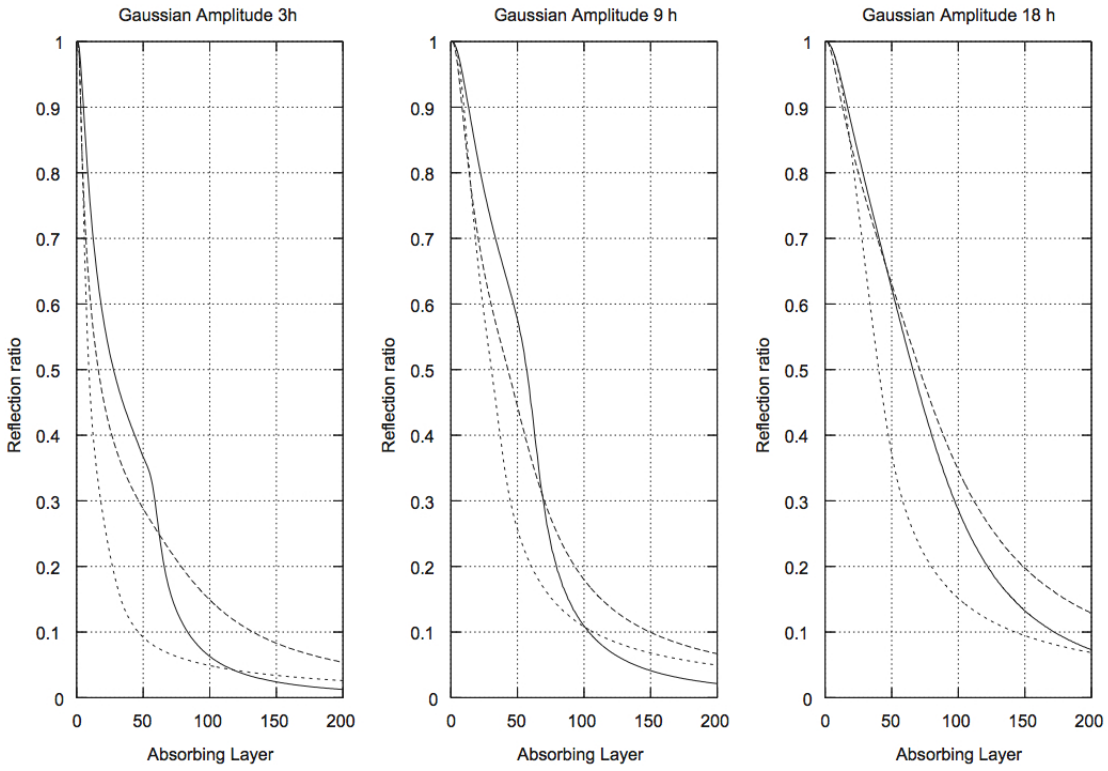
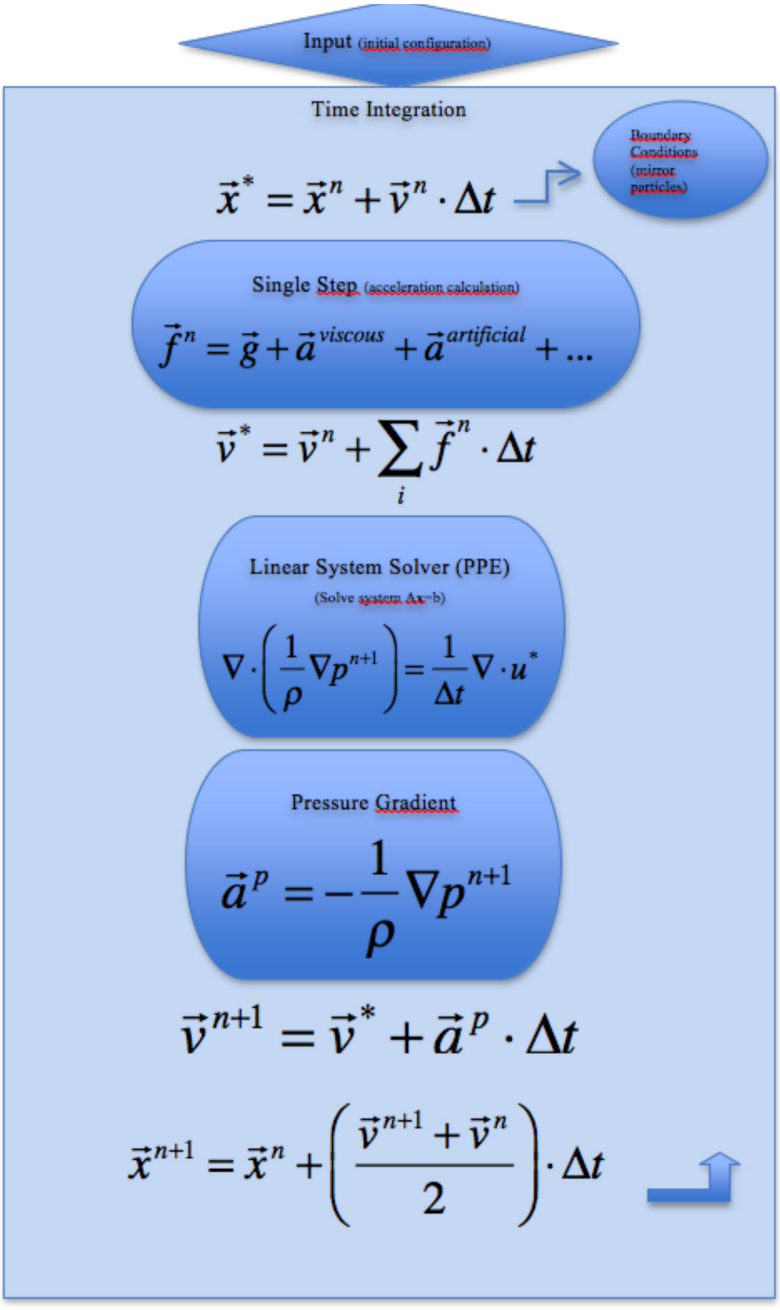


Figure 6.1: Reflection coefficients for increasing length of the damping layer. The results displayed with the solid, dashed and dotted lines are obtained with the clean hyperbolic damping function, f_1 pressure factor, f_1 and speed switch respectively.

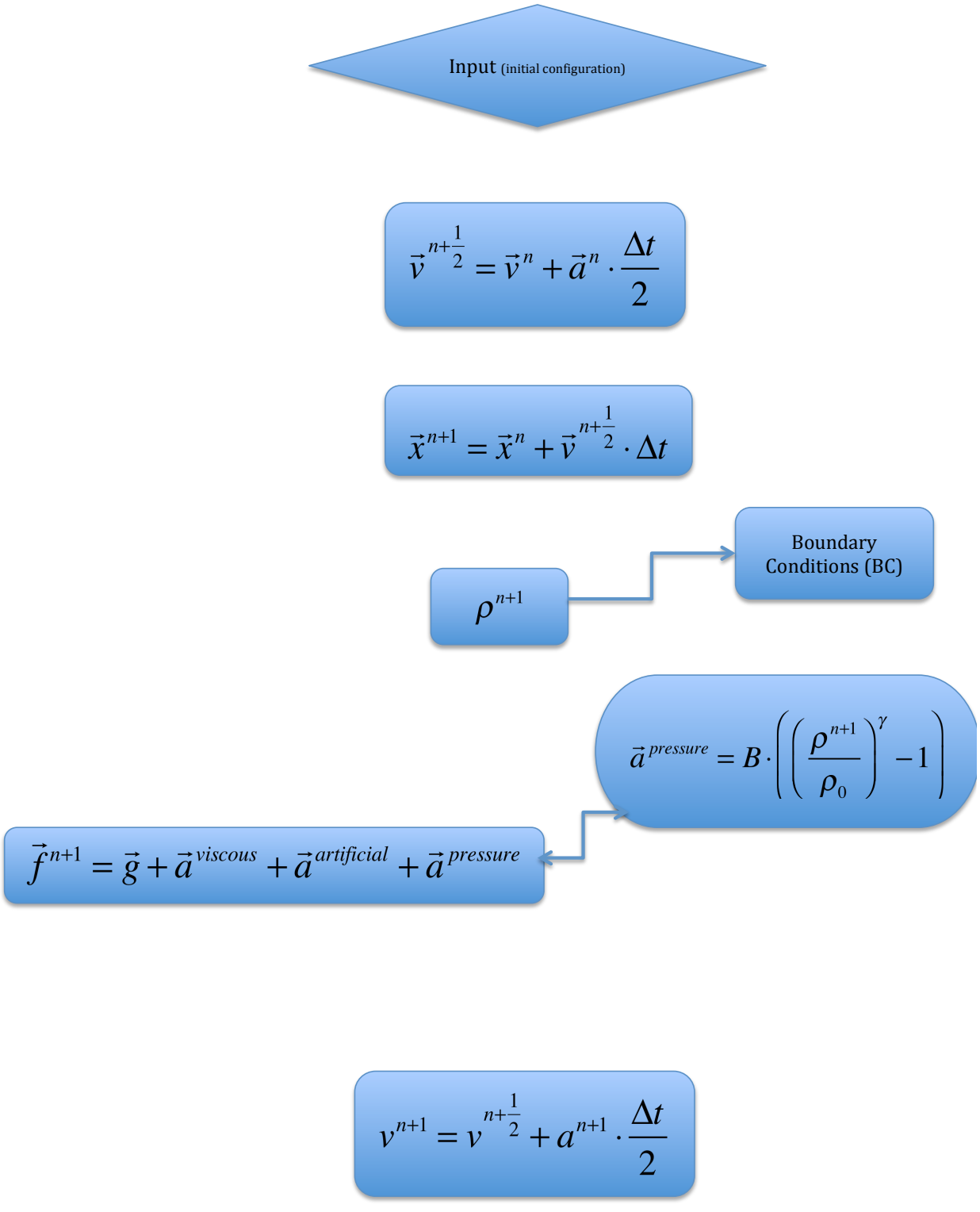
Appendix 1

Simplified block scheme of *ISPH* numerical code


Input (initial configuration)
Time Integration
Boundary Conditions (mirror particles)
$\vec{x}^* = \vec{x}^n + \vec{v}^n \cdot \Delta t$
Single Step (acceleration calculation)
$\vec{f}^n = \vec{g} + \vec{a}^{viscous} + \vec{a}^{artificial} + \ldots$
$\vec{v}^* = \vec{v}^n + \sum_i \vec{f}^n \cdot \Delta t$
Linear System Solver (PPE)
(Solve system Ax=b)
$\nabla \cdot \left(\frac{1}{\rho} \nabla p^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot u^*$
Pressure Gradient
$\vec{a}^p = -\frac{1}{\rho} \nabla p^{n+1}$
$\vec{v}^{n+1} = \vec{v}^* + \vec{a}^p \cdot \Delta t$
$\vec{x}^{n+1} = \vec{x}^n + \left(\frac{\vec{v}^{n+1} + \vec{v}^n}{2} \right) \cdot \Delta t$

Appendix 2

Simplified block scheme of *WCSPH* numerical code


Input (initial configuration)
$\vec{v}^{\,n+\frac{1}{2}} = \vec{v}^{\,n} + \vec{a}^{\,n} \cdot \frac{\Delta t}{2}$
$\vec{x}^{\,n+1} = \vec{x}^{\,n} + \vec{v}^{\,n+\frac{1}{2}} \cdot \Delta t$
ρ^{n+1}
Boundary Conditions (BC)
$\vec{a}^{\,pressure} = B \cdot \left(\left(\frac{\rho^{n+1}}{\rho_0} \right)^{\gamma} - 1 \right)$
$\vec{f}^{\,n+1} = \vec{g} + \vec{a}^{\,viscous} + \vec{a}^{\,artificial} + \vec{a}^{\,pressure}$
$v^{n+1} = v^{\,n+\frac{1}{2}} + a^{n+1} \cdot \frac{\Delta t}{2}$

References

S. Albensoeder and H.C. Kuhlmann. Accurate three-dimensional lid-driven cavity flow. *Journal of Computational Physics*, 206:536–558, 2005. vi, 38, 40

M. Antuono, A. Colagrossi, S. Marrone, and C. Lugni. Propagation of gravity waves through an SPH scheme with numerical diffusive terms. *Comput. Phys. Commun.*, 182(4):866–877, 2011. 58

A. Baker. Finite element computational fluid mechanics. *McGraw Hill*, 1985. 1

G. Batchelor. An introduction to fluid dynamics. *Cambridge University Press*, 1967. 3, 19

M. Becker and M. Teschner. Weakly compressible SPH for free surface flows. *Symposium on Comp. Animation*, pages 209–217, 2007. 3, 30

M. Becker, H. Tessendorf, and M. Teschner. Direct forcing for lagrangian rigid-fluid coupling. *EEE Transactions on Visualization and Comp. Graphics*, 15(3): 493–503, 2009. 3

J.B. Bell, P. Colella, and H.M. Glaz. A second-order projection method for the incompressible navier-stokes equations. *Journal of Comp. Phys.*, 85:257–283, 1989. 3

T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Comput. Methods Appl. Mech. Eng.*, 39(1):3–47, 1996. 2

W. Benz. Smoothed particle hydrodynamics: a review, in numerical modellying of non-linear stellar pulsation: problems and prospects. *Kluwer Academic, Boston*, 1989. 5

J.P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comp. Phys.*, 114:185–200, 1994. 57

J. Bonet and T.S.L. Lok. Variational and momentum preservation aspects of Smoothed Particle Hydrodynamics formulations. *Computat. Methods Appl. Mech. Engineering*, 180:97–115, 1999. 14

A.J. Chorin. Numerical solution of the navier stokes equations. *J. Math. Comput.*, 22:745–762, 1968. 3, 20

S.J. Cummins and M. Rudman. An SPH projection method. *J. Comput. Phys.*, 152:584–607, 1999. 3, 20

M. De Leffe, D. Le Touzé, and B. Alessandrini. SPH modeling of shallow-water coastal flows. *J. Hydraul. Res.*, 48:118–125, 2010. 59

B. Engquist and A. Majda. Absorbing boundary conditions for numerical simulation of waves. *Proc. Natl. Acad. Sci. USA*, 74(5):1765–1766, 1977. 57

D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183(1):83–116, 2002. 3

J. Fang and A. Parriaux. A regularized Lagrangian finite point method for the simulation of incompressible viscous flows. *Journal of Computational physics*, 227:8894–8908, 2008. 29

A. Fulk, D. A numerical analysis of smoothed particle hydrodynamics. *PhD. Thesis, Air Force Institute of Techonlogy*, 1994. 7

R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. astr. Soc.*, 181:375–389, 1977. 2, 13

L. Hernquist and N. Katz. TreeSPH: a unification of SPH with the hierarchical tree method. *The Astrophysical Supplement Series*, 70:419–446, 1989. 7

W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 139(1-4):556–593, 1981. 2

X.Y. Hu and N.A. Adams. An incompressible multi-phase SPH method. *Journal of Computational Physics*, 227:264–278, 2007. 22

S. Koshizuka and Y. Oka. Moving-Particle Semi-Implicit Methos for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, 123:421–434, 1996. 21, 44

M. Lastiwka, M. Basa, and N.J. Quinlan. Permeable and non-reflecting boundary conditions in SPH. *Int. J. Numer. Methods Fluids*, 61:709–724, 2009. 58

E.S. Lee. Truly incompressible approach for computing incompressible flow in SPH and comparisons with the traditional weakly compressible approach. *PhD Thesis. University of Manchester*, 2007. 26

E.S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Accuracy and stability in incompressible sph (ISPH) based on the projection method and a new approach. *Journal of Computational physics*, 228:6703–6725, 2009. 30

L.D. Libersky, A.G. Petscheck, T.C. Carney, J.R. Hipp, and F.A. Allahdadi. High strain lagrangian hydrodynamics: a three-dimensional SPH code for dynamic material response. *Journal of Computational Physics*, 109:67–75, 1993. 25

D.K. Lin, X.D. Li, and Q. Hu. Absorbing boundary condition for nonlinear euler equations in primitive variables based on the perfectly matched layer technique. *Comput. Fluids*, 40:333–337, 2011. 57

G.R. Liu. Meshfree methods: moving beyond the finite element method. *CRC Press, Boca Raton*, 2002. 2, 14

G.R. Liu and Y.T. Gu. An introduction to meshfree methods and their programming. *Springer, Dordrecht*, page 479, 2005. 2

G.R. Liu and M.B. Liu. Smoothed particle hydrodynamics: a meshfree particle method. *World Scientific*, 2003. 14

L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astron. J.*, 82:1013–1024, 1977. 2

P.A. Madsen and H.A. Shaffer. A discussion of artificial compressibility. *Coastal Eng.*, 53:93–98, 2006. 63

J.C. Martin and W.J Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos. Trans.of the Royal Soc. of London*, 244:312–324, 1952. vii, 44, 48, 49, 53, 54, 84

A. Modave, E. Deleersnijder, and J.M. Delhez. On the parameters of absorbing layers for shallow water models. *Ocean Dyn.*, 60:65–79, 2010. 57, 60, 61, 62

D. Molteni, A. Colagrossi, and G. Colicchio. On the use of an alternative water state equation for SPH. *Proc. of SPHERIC.*, pages 156–160, 2007. 19

J.J. Monaghan. On the problem of penetration in particle methods. *Journal of Computational physics*, 82:1–15, 1989. 27

J.J. Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30:543–574, 1992. 5, 7

J.J. Monaghan. Simulating free surface flows with SPH. *J. Comp. Phys.*, 110: 399–406, 1994. 3, 19, 24, 41

J.J. Monaghan. Sph without a tensile instability. *Journal of Computational physics*, 159:290–311, 2000. 28

J.J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703–1759, 2005. 14

J.J. Monaghan and R.A. Gingold. Shock simulation by the particle method of SPH. *Journal of Computational Physics*, 53:374–381, 1983. 17, 27

J.J. Monaghan and A. Kos. Solitary waves on a cretan beach. *ASCE J. of Waterways, Port, Coastal and Oc. Eng.*, 125(3):145–154, 1999. 24

J.J. Monaghan and J.C. Lattanzio. A refined particle method for astrophysical problems. *Astronomy and Astrophysics*, 149:135–143, 1985. 13, 28

J.J. Monaghan and J. Poinracic. Artificial viscosity for particle methods. *Applied Numerical Mathematics*, 1:187–194, 1985. 17

J.P. Morris, P.J. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using sph. *Journal of Computational physics*, 136:214–226, 1997. 44

M.W. Morton and D.F. Mayers. Numerical solution of partial differential equations, an introduction. *Cambridge University Press*, 2005. 1

C. Oger, M. Doring, B. Alessandrini, and P. Ferrant. An improved SPH method: towards higher order convergence. *Journal of Computational. Physics*, 225: 1472–1492, 2007. 11

T.J. Poinsot and S.K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *J. Comp. Phys.*, 101:104–129, 1992. 57

P.W. Randles and L.D. Libersky. Smoothed particle hydrodynamics some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 138:375–408, 1996. 25, 44

A. Riadh and S. Azzedine. A stabilized sph method for inviscid shallow water flows. *Int. J. Numer. Methods Fluids*, 47:139–159, 2005. 60

S. Shao and E. Lo. Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface. *Advances in Water Resources*, 26: 787–800, 2003. 21

E.F. Toro. Riemann solvers and numerical methods for fluid dynamics. *Springer-Verlag*, 1999. 1

R. Vacondio, B. Rogers, P. Stansby, and P. Mignosa. SPH modeling of shallow flow with open boundaries for practical flood simulation. *J. Hydraul. Eng.*, 138(6):530–541, 2012. 58

J. von Neumann and R.D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21:232–247, 1950. 18

H. Wendland. Computational aspects of radial basis function approximation. *Elsevier*, 1995. 13

R. Xu, D.L. Stansby, and D. Laurence. Accuracy and stability in incompressible sph (ISPH) based on the projection method and a new approach. *Journal of Computational physics*, 228:6703–6725, 2009. 22, 28, 44